

IMPLEMENTASI ALGORITME *MAPPING* DENGAN *BACKTRACKING* PADA *MOBILE ROBOT* DALAM *MAZE*

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun oleh:
Fikri Fauzan
NIM: 145150300111024



**PROGRAM STUDI TEKNIK KOMPUTER
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

IMPLEMENTASI ALGORITME MAPPING DENGAN BACKTRACKING PADA MOBILE ROBOT DALAM MAZE

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Teknik

Disusun Oleh :

Fikri Fauzan

NIM : 145150300111024

Skripsi ini telah diuji dan dinyatakan lulus pada
26 Desember 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Dahnia Syauqy, S.T., M.T., M.Sc.

NIK: 201607 870423 1 002

Dosen Pembimbing II



Rizal Maulana, S.T., M.T., M.Sc.

NIK: 201607 891009 1 001

Mengetahui

Ketua Jurusan Teknik Informatika




Tri Astoto Kurniawan, S.T., M.T., Ph.D

NIP: 19710518 200312 1 001

A

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata di dalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 13 Desember 2018



Fikri Fauzan

NIM: 145150300111024

KATA PENGANTAR

Dengan menyebut nama Allah Subhanahu Wa Ta'alla yang maha pengasih lagi maha penyayang, puji syukur saya panjatkan atas ke hadirat Allah Subhanahu Wa Ta'ala, karena ridha-Nya saya bisa menyelesaikan skripsi ini dan dengan menyelesaikan skripsi ini insya'Allah saya mendapatkan rahmat-Nya. Nabi Muhammad Shalallahu 'Alaihi Wasallam yang telah memberikan sebaik-baik contoh dalam meminta pertolongan kepada Allah Subhanahu Wa Ta'ala dan cara melewati segala kesulitan dengan ketenangan. Adapun maksud penyusunan skripsi ini adalah untuk memenuhi salah satu persyaratan dalam memenuhi ujian Sarjana Fakultas Ilmu Komputer dan mengharapkan rahmat Allah Subhanahu Wa Ta'ala. Judul skripsi yang disusun adalah: "Implementasi Algoritme *Mapping* Dengan *Backtracking* Pada *Mobile Robot* Dalam *Maze*".

Banyak cobaan dan hambatan yang saya alami dalam menyusun skripsi ini, akan tetapi semua itu teratasi dengan baik berkat ridha Allah Azza Wa Jalla, do'a orang tua yang sama seperti do'a seorang Nabi kepada umatnya dan dukungan dari banyak pihak. Berdasarkan hal tersebut, pada kesempatan ini saya mengucapkan terima kasih kepada:

1. Bapak Dahniyal Syauqy, S.T., M.T., M.Sc. selaku dosen pembimbing I yang telah banyak memberikan pengarahan dalam menyelesaikan laporan skripsi dan revisi-revisi yang diberikan sangat bermanfaat dalam keilmuan suatu penulisan.
2. Bapak Rizal Maulana, S.T., M.T., M.Sc. selaku Ketua Program Studi Teknik Komputer Universitas Brawijaya Malang dan selaku dosen pembimbing II yang telah banyak memberikan ide dalam penyelesaian skripsi, membimbing dengan sabar dan pengarahan yang sangat bermanfaat dalam penyelesaian skripsi.
3. Kedua orang tua yang telah meridhai saya, dengan ridhanya maka ridha Allah Subhanahu Wa Ta'ala hadir di setiap proses pengerjaan skripsi saya. Dan juga do'a kedua orang tua yang banyak berperan dalam penyelesaian skripsi.
4. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D. selaku Dekan Fakultas Ilmu Komputer Universitas Brawijaya Malang.
5. Bapak Heru Nurwasito, Ir., M.Kom. selaku Wakil Dekan I Bidang Akademik Fakultas Ilmu Komputer Universitas Brawijaya Malang.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D. selaku Ketua Jurusan Informatika Universitas Brawijaya Malang.
7. Seluruh dosen dan karyawan Fakultas Ilmu Komputer yang telah berperan dalam proses penyelesaian skripsi ini.
8. Pramanda Saputra yang membantu proses skripsi ini secara berkelompok.

9. Yusuf Hidayat yang membantu memberi saran dalam proses pengerjaan skripsi.

Saya menyadari bahwa laporan skripsi ini jauh dari sempurna, oleh karena itu untuk segala kritik dan saran yang membangun saya ucapkan terima kasih. Saya mengharapkan semoga laporan skripsi ini dapat berguna.

Malang, 13 Desember 2018

Penulis
Fikrifauzan17@gmail.com



ABSTRAK

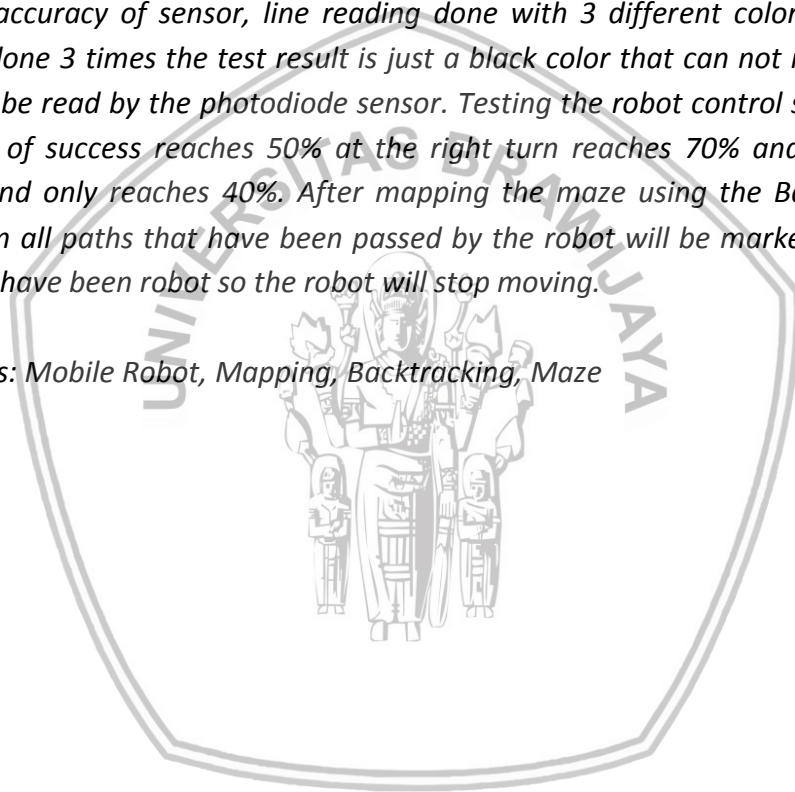
Autonomous Mobile Robot marak dikembangkan dalam banyak penelitian terutama robot *Maze Wall Follower* yang dapat mencari jalan keluar dari sebuah labirin yang sangat rumit. Pada penelitian ini *Robot Maze Wall Follower* menggunakan Algoritme *Backtracking* yaitu menelusuri seluruh kemungkinan jalur yang ada pada *maze* dan mengenali setiap node yang dilewati agar tidak mengulangi jalur yang sudah dilewati. *Prototype* ini dibangun menggunakan beberapa modul yaitu Arduino NANO sebagai pemrosesannya, sensor Ultrasonik, sensor Photodiode dan L298N Driver Motor. Hasil pengujian dari pengukuran jarak dinding menggunakan sensor Ultrasonik HC SR04 yang dilakukan pengujian sebanyak 4 kali dengan jarak yang berbeda-beda yang didapatkan semakin jauh jarak sensor ke dinding semakin kecil juga tingkat keakuratan sensor, pembacaan garis dilakukan dengan 3 warna berbeda dan setiap warna dilakukan 3 kali pengujian hasilnya hanya warna hitam yang tidak dapat memantulkan cahaya yang dapat dibaca oleh sensor photodiode. Pengujian sistem kendali robot, pada belokan kiri keberhasilan mencapai 50% pada belokan kanan mencapai 70% dan ketika putar balik hanya mencapai 40%. Setelah melakukan pemetaan *maze* menggunakan Algoritme *Backtracking* semua jalur yang telah dilewati robot akan ditandai, setelah semua jalur itu sudah tertanda robot maka robot akan berhenti bergerak.

Kata kunci: *Mobile Robot, Mapping, Backtracking, Maze*

ABSTRACT

Autonomous Mobile Robot is rife in many researches especially the Maze Wall Follower robot that can find its way out of a very complicated labyrinth. In this study the Robot Maze Wall Follower using Backtracking algorithm is to trace all possible paths in the maze and recognize each node that is passed so as not to repeat the path that has been passed. This prototype is built using several modules of Arduino NANO as its processing, Ultrasonic sensor, Photodiode sensor and L298N Driver Motor. Test results from measurement of wall distance using HC SR04 Ultrasonic sensor which is tested as much as 4 times with different distance obtained by the distance of sensor distance to wall the smaller also the level of accuracy of sensor, line reading done with 3 different color and every color is done 3 times the test result is just a black color that can not reflect light that can be read by the photodiode sensor. Testing the robot control system, the left turn of success reaches 50% at the right turn reaches 70% and when the turnaround only reaches 40%. After mapping the maze using the Backtracking algorithm all paths that have been passed by the robot will be marked, after all the lines have been robot so the robot will stop moving.

Keywords: Mobile Robot, Mapping, Backtracking, Maze



DAFTAR ISI

PENGESAHAN	ii
PERNYATAAN ORISINALITAS	iii
KATA PENGANTAR.....	iv
ABSTRAK.....	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR.....	xi
DAFTAR TABEL.....	xiii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan	2
1.4 Manfaat.....	2
1.5 Batasan Masalah	3
1.6 Sistematika Pembahasan	3
BAB II TINJAUAN PUSTAKA.....	5
2.1 Landasan Kepustakaan.....	5
2.2 Dasar Teori	6
2.2.1 <i>Mobile Robot</i>	6
2.2.2 Arduino NANO.....	8
2.2.3 Sensor Ultrasonik	10
2.2.4 Driver Motor L298N	12
2.2.5 Motor DC.....	13
2.2.6 <i>Power Supply</i>	14
2.2.7 Sensor Garis TCRT5000	15
2.2.8 Algoritme Mapping dengan <i>Backtracking</i>	17
2.2.9 Arduino IDE	18
2.2.10 EEPROM	20
BAB III METODOLOGI PENELITIAN	21
3.1 Pre Riset atau Studi Literatur	22

3.2	Analisis Kebutuhan Sistem	22
3.2.1	Kebutuhan Fungsional	23
3.2.2	Kebutuhan Non-Fungsional	23
3.3	Perancangan Sistem.....	23
3.4	Implementasi	24
3.5	Pengujian dan Analisis	24
3.6	Pengambilan Kesimpulan dan Saran.....	24
BAB IV REKAYASA KEBUTUHAN		25
4.1	Gambaran Umum Sistem.....	25
4.2	Analisis Kebutuhan Sistem.....	25
4.2.1	Kebutuhan Fungsional	25
4.2.2	Kebutuhan Non-Fungsional	27
4.3	Batasan Desain Sistem	28
BAB V PERANCANGAN DAN IMPLEMENTASI		29
5.1	Perancangan Sistem.....	29
5.1.1	Gambaran Umum Sistem.....	29
5.1.2	Perancangan <i>Prototype</i> Alat	29
5.1.3	Perancangan Perangkat Keras	30
5.1.4	Perancangan Perangkat Lunak.....	33
5.2	Implementasi	41
5.2.1	Implementasi <i>Prototype</i> Alat	41
5.2.2	Implementasi Perangkat Keras	42
5.2.3	Implementasi Perangkat Lunak.....	43
BAB VI PENGUJIAN DAN ANALISIS		49
6.1	Pengujian Pembacaan Sensor Ultrasonik HC-SR04.....	49
6.1.1	Tujuan Pengujian.....	49
6.1.2	Prosedur Pengujian	49
6.1.3	Hasil dan Analisis Pengujian	50
6.2	Pengujian Pembacaan Modul Sensor Photodiode.....	51
6.2.1	Tujuan Pengujian.....	51
6.2.2	Prosedur Pengujian	51
6.2.3	Hasil dan Analisis Pengujian	53
6.3	Pengujian Sistem Kendali Robot	53

6.3.1	Tujuan Pengujian.....	54
6.3.2	Prosedur Pengujian	54
6.3.3	Hasil dan Analisis Pengujian.....	54
6.4	Pengujian Mapping dengan Algoritme <i>Backtracking</i>	55
6.4.1	Tujuan Pengujian.....	55
6.4.2	Prosedur Pengujian	55
6.4.3	Hasil dan Analisis Pengujian.....	55
BAB VII PENUTUP		58
7.1	Kesimpulan.....	58
7.2	Saran	59
DAFTAR PUSTAKA.....		60



DAFTAR GAMBAR

Gambar 2. 1 Diagram blok system	6
Gambar 2. 2 Mobile Robot.....	7
Gambar 2. 4 Arduino	8
Gambar 2. 4 Sensor Ultrasonik	10
Gambar 2. 5 Skematik Sensor Ultrasonik.....	10
Gambar 2. 6 Diagram Clock Sensor UltrasonikHC-SR04	11
Gambar 2. 8 Motor Driver L298N	12
Gambar 2. 9 Konstruksi Pin Driver Motor.....	12
Gambar 2. 10 Simbol dan Bentuk Motor DC.....	14
Gambar 2. 11 12 Volt Baterai.....	14
Gambar 2. 12 Sensor Garis TCRT5000	15
Gambar 2. 13 Sensor photodiode tidak terkena pantulan cahaya.....	15
Gambar 2. 14 Sensor photodiode terkena pantulan cahaya	16
Gambar 2. 15 Ilustrasi mekanisme sensor garis	16
Gambar 2. 16 Pohon Solusi Algoritme Backtracking	17
Gambar 2. 17 Sketch Tampilan Arduino IDE.....	18
Gambar 2. 18 Bagian-bagian Arduino IDE.....	19
Gambar 3. 1 Diagram Alir Metodologi Penelitian.....	21
Gambar 3. 2 Analisis Kebutuhan	22
Gambar 5. 1 Diagram Blok Sistem.....	30
Gambar 5. 2 Desain Prototype Robot Tampak Atas	31
Gambar 5. 3 Desain Prototype Robot Tampak Bawah	31
Gambar 5. 4 Konfigurasi Pin Sensor Ultrasonik	32
Gambar 5. 5 Konfigurasi Pin Driver Motor DC L298N.....	33
Gambar 5. 6 Konfigurasi Pin Sensor Photodiode	34
Gambar 5. 7 Flowchart Pembacaan Sensor Ultrasonik	35
Gambar 5. 8 Flowchart Sensor Photodiode	36
Gambar 5. 9 Flowchart L298N Driver Motor	37
Gambar 5. 10 Follow Right.....	38
Gambar 5. 11 Rancangan maze yang digunakan	39
Gambar 5. 12 Variabel pada Setiap Array.....	39
Gambar 5. 13 Penomoran Grid pada Array	40
Gambar 5. 14 Flowchart Algoritme Mapping dengan Input Sensor Photodiode .	41
Gambar 5. 15 Robot Bagian Atas	42
Gambar 5. 16 Robot Bagian Bawah	43
Gambar 5. 17 Robot Bagian Belakang.....	44

Gambar 5. 18 Robot Bagian Belakang.....	44
Gambar 6. 1 Grafik Pengujian dan Analisis	49
Gambar 6. 2 Pengujian Sensor Ultrasonik	50
Gambar 6. 3 Pegujian Pada Warna Merah.....	52
Gambar 6. 4 Pengujian Pada Warna Putih.....	52
Gambar 6. 5 Pengujian Pada Warna Hitam	53
Gambar 6. 6 Pengujian Pergerakan Robot.....	54
Gambar 6. 7 Arah Jalannya Robot dan Hasil Mapping	56
Gambar 6. 8 Tampilan Data Hasil Mapping pada Serial Monitor	56



DAFTAR TABEL

Tabel 2. 1 Spesifikasi Arduino NANO	8
Tabel 2. 2 Fungsi Pin-pin Sensor Ultrasonik	11
Tabel 5. 1 Konfigurasi Pin Sensor Ultrasonik	31
Tabel 5. 2 Konfigurasi Pin Driver Motor DC L298N	32
Tabel 5. 4 Pembacaan Sensor Ultrasonik	43
Tabel 5. 5 Pembacaan Modul Sensor Photodiode	44
Tabel 5. 6 Pembacaan IC Driver Motor L298N	45
Tabel 5. 7 Pergerakan Follow Right	46
Tabel 5. 9 Penyimpanan ke Memori EEPROM dan Maze Mapping Robot	47
Tabel 6. 1 Hasil Pengukuran Jarak Menggunakan Sensor Ultrasonik	50
Tabel 6.2 Hasil Pengujian Sensor Photodiode	53
Tabel 6. 3 Hasil Pengujian Pergerakan Robot	54



BAB I PENDAHULUAN

1.1 Latar Belakang

Mobile robot menjadi salah satu yang belakangan ini menarik minat para peneliti untuk dikembangkan. Suatu robot yang dapat bergerak sendiri dari suatu tempat lain disebut *mobile robot* (Hartanto, 2005). Fungsi *mobile robot* dapat meniru dari fungsi makhluk hidup yang dapat berpindah dari satu tempat ke tempat lain, robot ini dirancang untuk berbagai keperluan seperti transportasi, mengangkut barang, alat untuk hiburan dan lain-lain.

Sejak robot dibuat dapat melakukan tugas-tugas secara mandiri, bidang *Autonomous Mobile Robot* semakin marak dikembangkan dalam banyak penelitian (Isfan Fauzi, 2017). Meskipun robot dengan sensor berbasis citra menjadi tren pada saat ini, namun robot penghindar halangan masih banyak digunakan dalam berbagai penelitian. Robot penghindar banyak digunakan di dalam berbagai penelitian karena robot penghindar banyak dipakai oleh pabrik dan industri lain yang berhubungan dengan transportasi barang. Robot penghindar halangan dirancang karena mempunyai desain rangkaian yang mudah dirangkai dan memiliki *cost* yang terjangkau. Robot penghindar halangan ini juga dapat dikembangkan menjadi robot *maze solver* dengan ditambahkan Algoritme kedalam robot tersebut. Robot *maze solver* adalah robot yang dapat menemukan jalur keluar dari sebuah labirin dengan cepat dan tepat.

Mobile robot merupakan sebuah rangkaian robot yang mempunyai ciri-ciri yang khas yaitu mempunyai penggerak berupa motor untuk menggerakkan seluruh tubuh robot tersebut, sehingga robot dapat bergerak dari tempat satu ketempat lain. *Mobile robot* saat banyak digunakan untuk kontes robot *micromouse*. Robot *micromouse* adalah robot yang dapat bergerak di dalam sebuah *maze* tanpa menyentuh dinding labirin, robot dapat menentukan ke arah mana robot dapat menentukan jalan keluar dari labirin.

Robot *micromouse* mempunyai 2 tipe yaitu dalam *solving maze* dan tipe *mapping*. Tipe *solving* robot akan melalui semua jalur yang ada pada *maze*, sedangkan tipe *mapping* robot akan bergerak sesuai dengan data yang tersimpan pada memori pada *microcontroller* yang di dapatkan dari tipe eksplorasi. Bahasa pemrograman yang digunakan adalah bahasa C dan *microcontroller* yang biasanya digunakan adalah sebuah modul *microcontroller* 168 yang biasa disebut *Arduino* serta Algoritme yang dipakai adalah *backtracking*. Algoritme *Backtracking* adalah Algoritme yang berdasar pada Algoritme DFS (*Deep Field Search*) untuk menelusuri semua solusi permasalahan lebih mangkus dari Algoritme lain. *Backtracking* adalah Algoritme yang lebih kompleks dari Algoritme *Brute-Force*, kedua algoritma tersebut bekerja secara sistematis dalam mencari solusi dari

masalah yang memiliki solusi yang kompleks (D.H. Lehmer, 1950). Prinsip dasarnya adalah mencoba semua kemungkinan yang ada untuk mencari solusi.

Pada penelitian yang akan dilakukan, proyek ini menggunakan perangkat Arduino Nano Atmega 168. Arduino merupakan mikrokontroler yang dirancang untuk mengembangkan objek, mengambil dan memasukan atau input pada sensor dan dapat mengontrol berbagai jenis aktuator ataupun rangkaian output lainnya. *Robot ini* dapat menggunakan Algoritme *mapping* dengan *backtracking* yang diterapkan pada eksplorasi *maze*. Pada sistem yang ada pada *mobile robot*, peneliti menggunakan sensor ultrasonik yang berfungsi sebagai pendeteksi halangan tembok yang terdapat pada *maze*. Kemudian *Mobile robot* menentukan belokan setiap persimpangan dan arah jalur yang dilalui *Mobile robot* dan mengolahnya dengan metode *backtracking* untuk eksplorasi map yang ada pada *maze*.

1.2 Rumusan Masalah

Berikut adalah rumusan masalah yang didapatkan yang bertujuan mengembangkan proyek ini :

1. Bagaimana robot mendeteksi objek dinding dan garis hitam pada setiap node sebagai awalan untuk menerapkan Algoritme *Backtracking* ?
2. Bagaimana cara mengenali jalur setiap node dengan menggunakan sensor garis TCRT5000?
3. Bagaimana akurasi Algoritma *Backtracking* dapat menemukan semua jalur yang ada pada sebuah labirin tanpa mengulangi jalur sebelumnya?

1.3 Tujuan

Dengan perumusan masalah yang ada, didapatkanlah tujuan yang berkaitan dengan perumusan masalah yang ada:

1. Robot dapat mendeteksi dinding dan garis hitam pada setiap node sebagai awalan untuk menerapkan Algoritme *Backtracking*.
2. Dapat mengenali setiap node yang telah ditempuh dengan menggunakan sensor garis TCRT5000
3. Robot dapat menemukan semua jalur yang ada pada sebuah labirin dan tidak melewati jalur yang sebelumnya sudah dilalui.

1.4 Manfaat

Manfaat yang bisa didapat dari fungsi sistem yang dilihat, penulis mendapatkan manfaat sebagai berikut :

1. Membantu perancang *micromouse* dengan tipe *explore* merancang *Mobile robot* yang dapat mengeksplorasi keseluruhan *map* pada *maze*.

2. Terealisasinya robot *micromouse* menggunakan sensor ultrasonik yang dapat mengexplore seluruh map pada *maze*.
3. Mendapat keuntungan ilmu serta kebanggan bagi diri sendiri serta dapat membantu para ahli dalam penelitiannya.

1.5 Batasan Masalah

Agar pembahasan menjadi terarah sesuai dengan rumusan masalah dan latar belakang yang telah dibahas. Karena itu pembatasan sebuah penelitian sangat diperelukan. Berikut merupakan batasan masalah yang di dapatkan dari permasalahan yang ada:

1. Robot ini hanya digunakan di daerah dengan medan kering.
2. Jangkauan jarak sensor dengan halangan pada *maze* yang dapat dibaca.
3. Permukaan *maze* yang datar dan tidak bergelombang.
4. Setiap node pada *maze* dibatasi garis hitam.

1.6 Sistematika Pembahasan

Untuk menjelaskan secara singkat tentang isi dari skripsi ini, dengan mengelompokkan materi menjadi beberapa sub bab :

BAB I : PENDAHULUAN

Penjelasan secara umum tentang informasi yang berkaitan dengan skripsi ini dengan judul "*Implementasi Algoritme Mapping Dengan Backtracking Pada Mobile Robot Dalam Maze*"

BAB II : LANDASAN KEPUSTAKAAN

Bab ini mendeskripsikan landasan kepastakaan dari tugas akhir ini yang akan dibuat dengan judul "*Implementasi Algoritme Mapping Dengan Backtracking Pada Mobile Robot Dalam Maze*". Bab ini juga menjelaskan konsep dasar Algoritme *backtracking*, Arduino Nano, *power supply*, driver motor dc dan motor dc pada rangkaian alat, dan informasi yang berkaitan dengan batasan masalah yang didapatkan peneliti.

BAB III : METODE PENELITIAN

Bab ini menjelaskan prinsip kerja serta, alur yang dipakai dalam peneliian ini. Alur yang digunakan adalah pre-riset atau studi literatur, analisis kebutuhan sistem yang berisi kebutuhan fungsional dan non-fungsional, perancangan sistem seperti perancangan alat dan perangkat lunak, implemantasi algorima *backtracking*, pengujian dan analisis sistem serta cara pengambilan kesimpulan dan saran.

BAB IV : REKAYASA KEBUTUHAN

Bab ini menjelaskan mengenai rekayasa kebutuhan yang terdiri dari kebutuhan fungsional dan non-fungsional dalam membangun prototype alat *Mobile robot* pada kontes *micromouse* dengan tipe robot explorasi *maze* menggunakan Algoritme *backtracking*, kebutuhan alat dan perangkat lunak.

BAB V : PERANCANGAN DAN IMPLEMENTASI

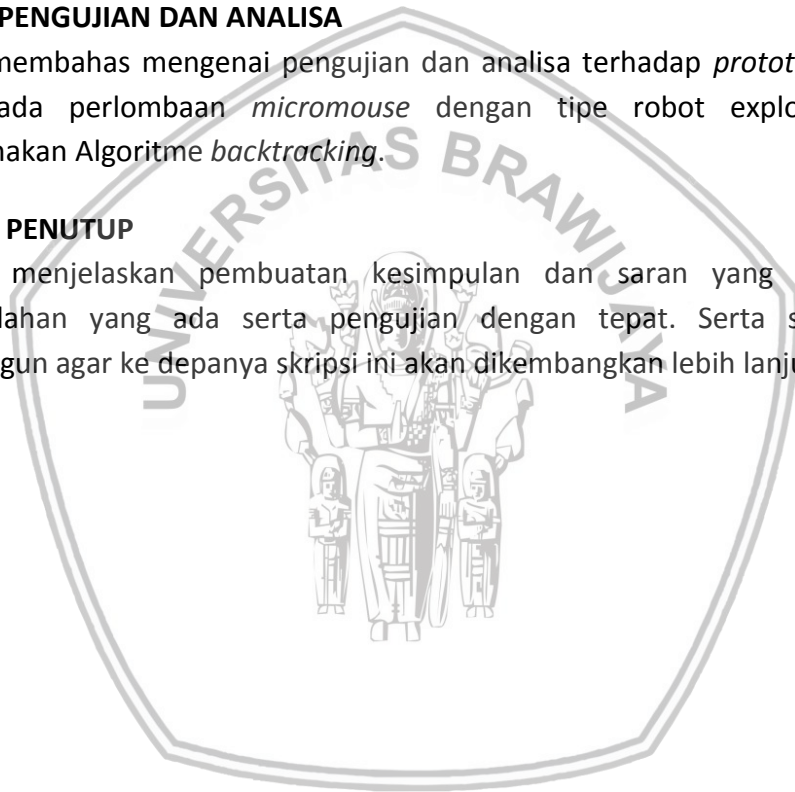
Bab ini menjelaskan mengenai perancangan prototype robot pada *Mobile robot* pada perlombaan *micromouse* dengan tipe robot *solver maze* menggunakan Algoritme *backtracking*.

BAB VI : PENGUJIAN DAN ANALISA

Bab ini membahas mengenai pengujian dan analisa terhadap *prototype Mobile robot* pada perlombaan *micromouse* dengan tipe robot explorasi *maze* menggunakan Algoritme *backtracking*.

BAB VII : PENUTUP

Bab ini menjelaskan pembuatan kesimpulan dan saran yang didapatkan permasalahan yang ada serta pengujian dengan tepat. Serta saran yang membangun agar ke depannya skripsi ini akan dikembangkan lebih lanjut.



BAB II TINJAUAN PUSTAKA

2.1 Landasan Kepustakaan

Setelah peneliti melakukan penelitian terhadap beberapa bahan yang akan diteliti, ada beberapa yang berkaitan dengan penelitian yang akan dilakukan peneliti.

Menurut jurnal yang berjudul "*Maze Solving Mobile Robot*". Robot *mobile* adalah Kombinasi perangkat, *servomotor* dan sensor, di bawah sistem kontrol yang beroperasi di ruang nyata, dan dirancang untuk melakukan operasi penyeberangan sebuah *maze*, terlepas dari bentuknya, dengan mengolah informasi yang diterima dari sensor (Gelu-Ovidiu, 2015).

Masalah robot *maze* merupakan bidang robotika yang penting dan didasarkan pada Algoritme pengambilan keputusan. Ini memerlukan analisis lengkap tentang ruang kerja atau *maze* dan perencanaan yang tepat. Menurut jurnal yang berjudul "*Design and Implementation of a Path Finding Robot Using FloodFill Algorithm*". Algoritme *floodfill* dan modifikasi memori yang diakses digunakan secara luas untuk masalah robot *maze*. Algoritme *floodfill* memberi nilai pada setiap node yang mewakili jarak simpul dari pusat. *floodfill* Algoritme *maze* ketika robot mencapai sel baru atau node (Tjiharjadi dan Semuil, 2016).

Sedangkan menurut dalam jurnal yang berjudul "*Autonomous Multi-mobile Robot Sistem: Simulasi dan Implementasi menggunakan Fuzzy Logic*". Dengan menggunakan pendekatan logika *fuzzy*. Di lingkungan robot multi-mobile otonom, perencanaan jalur dan penghindaran tabrakan Adalah fungsi penting yang digunakan untuk melakukan suatu tugas secara kolaboratif dan kooperatif. Studi ini mempertimbangkan Masalah penting dan menantang ini. Pendekatan yang diusulkan didasarkan pada bidang potensial Metode dan sistem logika *fuzzy*. Pertama, perencana jalur global memilih jalur robot yang meminimalkan Nilai potensial dari masing-masing robot ke targetnya sendiri menggunakan medan potensial. Lalu, seorang perencana jalur lokal Memodifikasi jalur dan orientasi dari perencana global untuk menghindari tabrakan dengan statis dan dinamis Hambatan menggunakan sistem logika *fuzzy* (Kim, dkk, 2013).

Dari ketiga penelitian diatas dapat disimpulkan bahwa Persamaan dengan penelitian yang diambil penulis merupakan suatu gabungan dari ketiga penelitian diatas, yaitu sama-sama membuat sistem keamanan menggunakan *Mobile robot*, yang tujuannya adalah membuat rancangan robot eksplorasi untuk daerah berbahaya menggunakan Algoritme *backtracking* dengan *floodfill*. Perbedaanya dengan penelitian sebelumnya adalah objek, tema yang diangkat dari penulis berbeda dengan penelitian yang sebelumnya dalam penelitian ini penulis

mengangkat tema rancangan robot eksplorasi untuk daerah berbahaya menggunakan metode *backtracking algorithm*.

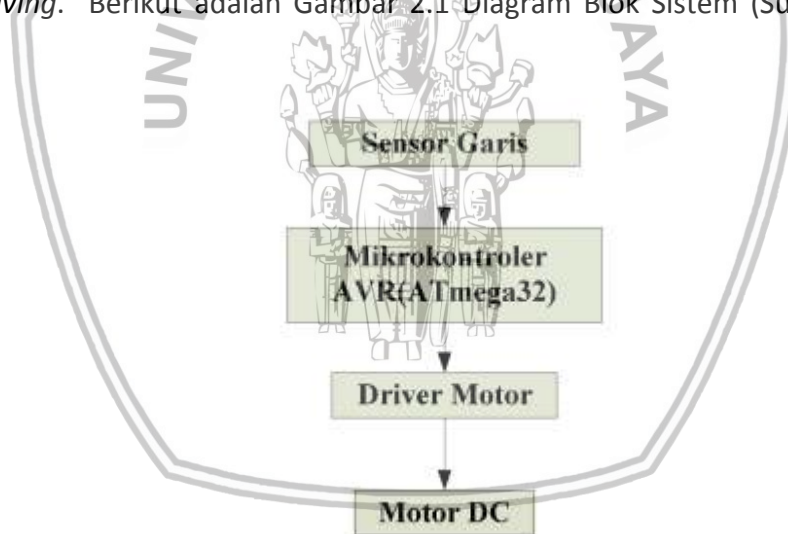
2.2 Dasar Teori

2.2.1 Mobile Robot

Robot merupakan kumpulan dari perangkat mekanik yang digabungkan dan dapat dikedalikan ataupun diprogram agar dapat bergerak sendiri. Salah satu tipe robot yang banyak dikembangkan adalah mobile robot. Mobile robot adalah robot yang diprogram agar dapat bergerak dari suatu tempat ke tempat lain. Ada macam-macam mobile robot yang biasa digunakan adalah robot beroda dan robot berkaki. Kebanyakan para peneliti lebih memilih membuat robot beroda karena perangkatnya lebih mudah dan murah.

Ada dua tipe mobile robot yaitu robot darat dan udara. Robot udara contohnya drone sedangkan robot darat contohnya *Micromouse*. Robot udara biasanya digunakan untuk mengintai daerah-daerah yang berbahaya sedangkan robot darat digunakan untuk mengangkut barang.

Contoh lainnya yang akan dibahas adalah Mobile Robot untuk melakukan *maze solving*. Berikut adalah Gambar 2.1 Diagram Blok Sistem (Supani Ahyar, 2011):



Gambar 2. 1 Diagram Blok Sistem

Fungsi dan cara kerja masing-masing blok sistem :

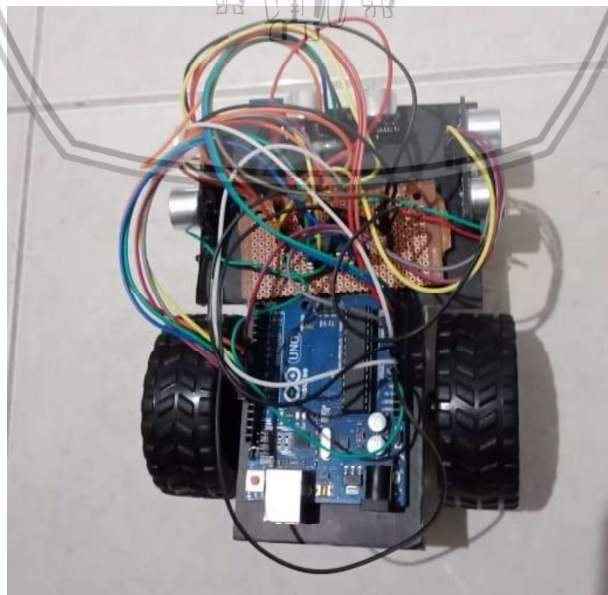
1. Blok Sensor Garis: blok ini berfungsi mendeteksi garis yang berwarna hitam atau putih. Pada blok ini terdiri dari 2 komponen yang berfungsi sebagai pemancar dan penerima. Komponen yang berfungsi sebagai pemancar yaitu led super bright, dan komponen yang berfungsi sebagai penerima yaitu phototransistor. Cara kerjanya yaitu led akan di pancarkan kegaris hitam atau putih selanjutnya cahaya yang dipantulkan tersebut akan diterima oleh phototransistor, ketika led dipancarkan

kegaris hitam maka cahaya akan diserap sedangkan pada garis putih cahaya akan dipantulkan.

2. Blok Mikrokontroler: blok ini berfungsi sebagai kendali yang akan mengatur jalannya mobile robot. Cara kerjanya yaitu mikrokontroler akan mendapatkan data dari sensor garis berupa sinyal analog selanjutnya data tersebut akan dikonversi ke sinyal digital melalui fitur ADC (Analog to Digital Converter) internal yang terdapat di mikrokontroler. Setelah dikonversi ke sinyal digital, mikrokontroler akan memerintahkan driver motor untuk menggerakkan motor DC melalui Bahasa pemrograman tingkat tinggi yang menyebabkan robot bergerak.
3. Blok Driver Motor: blok ini berfungsi untuk mengatur arah putaran motor DC, baik secara satu arah ataupun dua arah/bolak-balik. Cara kerjanya yaitu motor dc akan berputar searah jarum jam ataupun kebalikannya tergantung dari instruksi yang diberikan mikrokontroler dengan cara memberikan logika 1 atau 0 pada pin IC driver motor.
4. Blok Motor DC: blok ini berfungsi untuk memutar roda pada mobile robot.

Perangkat hardware pada Gambar 2.2 (perangkat keras) yang dipakai pada system mobile robot *solving maze*, antara lain:

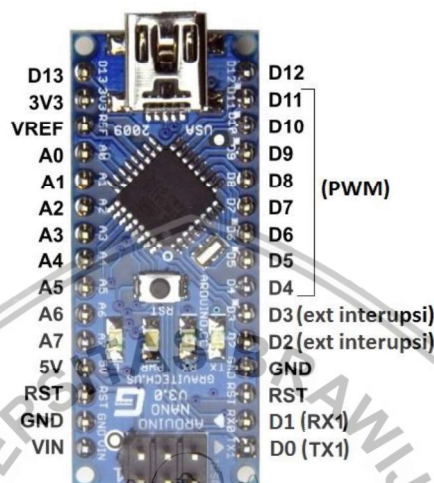
1. Mekanik Robot
2. Arduino Nano
3. Sensor garis
4. Sensor Ultrasonik
5. Motor DC



Gambar 2. 2 Mobile Robot

2.2.2 Arduino NANO

Arduino adalah pengendali mikro single-board yang bersifat *open-source*, diturunkan dari *Wiring platform*, dirancang untuk memudahkan penggunaan elektronik dalam berbagai bidang. *Hardware*nya memiliki prosesor *Atmel AVR* dan *software*nya memiliki bahasa pemrograman sendiri. Berikut merupakan tampilan dari Arduino NANO Gambar 2.3.



Gambar 2. 3 Arduino
(Sumber: Arduino, 2017)

Saat ini Arduino sangat populer di seluruh dunia. Banyak pemula yang belajar mengenal robotika dan elektronika lewat Arduino karena mudah dipelajari. Tapi tidak hanya pemula, para hobbyist atau profesional pun ikut senang mengembangkan aplikasi elektronik menggunakan Arduino. Bahasa yang dipakai dalam Arduino bukan *assembler* yang relatif sulit, tetapi bahasa C yang disederhanakan dengan bantuan pustaka-pustaka (*libraries*) Arduino (ecadio.com, 2018). Tabel 2.1 merupakan spesifikasi Arduino Nano.

Tabel 2. 1 Spesifikasi Arduino NANO

Chip mikrokontroller	Atmega168
Tegangan operasi	5V
Tegangan input (yang direkomendasikan, via jack DC)	7V - 12V
Tegangan input (limit, via jack DC)	6V - 20V
Digital I/O pin	14 buah, 6 diantaranya menyediakan PWM
Analog Input pin	8 buah
Arus DC per pin I/O	20 mA
Arus DC pin 3.3V	40 mA
Memori Flash	16 KB, 2 KB telah digunakan untuk bootloader

SRAM	2 KB
EEPROM	1 KB
<i>Clock speed</i>	16 Mhz
Dimensi	4.5 mm x 18 mm
Berat	5 g

(Sumber: ecadio.com, 2018)

A. Daya

Pada Arduino NANO menggunakan daya yang masuk melalui USB port atau dapat juga menggunakan pin tegangan Vin atau 5v dan GND untuk tegangan kerja 7 sampai 12v.

B. Memori

Mikrokontroler Atmega 168 dilengkapi dengan flash memori 16kbyte digunakan sebagai menyimpan kode program utama. Pada flash memori tersebut telah terpakai 2kb untuk bootloader. Atmega 168 juga dilengkapi memori SRAM dan EEPROM. SRAM digunakan untuk penyimpanan selama program utama bekerja. Besar memori sebesar 1kb sedangkan untuk EEPROM digunakan untuk menyimpan data secara permanen. Besar EEPROM sebesar 512b.

C. Input dan Output

Pada I/O Arduino NANO terdapat 14pin digital diantara 14 pin tersebut ada 8pin terdapat fungsi PWM (*pulse width modulation*) diantaranya D4, D5, D6, D7, D8, D9, D10, D11. Pin digital mengeluarkan tegangan 5v untuk mode HIGH (logika 1) dan 0V untuk mode LOW (logika 0) jika pin sebagai output. Arus listrik yang dapat dilewati pin digital I/O adalah 40mA resistor *pull-up* sebesar 20 -50 kOhm. Pin D0 dan D1 berfungsi sebagai TX dan RX untuk komunikasi data. Pin D2 dan D3 sebagai interupsi pemicu eksternal. Interupsi terjadi jika timbul kenaikan atau penurunan tegangan pada pin tersebut. Pin D10, D11, D12, D13 untuk komunikasi SPI dan pin D13 terhubung ke LED.

Terdapat 8 pin analog dalam Arduino NANO. Setiap pin analog terhubung ke ADC (*analog to digital converter*). Pada pin analog dapat digunakan sebagai I/O digital kecuali pin A6 dan A7 yang hanya dapat digunakan dalam I/O data Analog saja.

D. Komunikasi

Atmega 168 dilengkapi dengan komunikasi serial UART TTL, yang ada pada pin D0 dan D1. Modul juga dilengkapi sebuah IC FTDI 232 RI yang dapat terhubung langsung ke komputer dan menghasilkan virtual com-port pada sistem operasi.

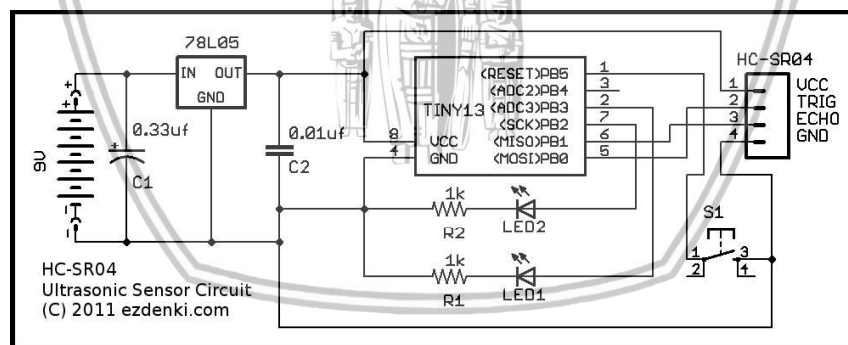
2.2.3 Sensor Ultrasonik

Sensor ultrasonik merupakan sebuah sensor yang berfungsi untuk mendeteksi jarak dari sensor ke objek yang dipantulkan gelombang suara. Cara kerjanya sensor mengeluarkan gelombang dan setelah dipantulkan kembali gelombang tersebut akan kembali masuk ke sensor. Dari pantulan gelombang itu terdapat interval waktu antara gelombang dikeluarkan dan diterima kembali. Dari interval tersebut sensor dapat menghitung jarak suatu objek. Gambar 2.4 merupakan tampilan dari depan sensor Ultrasonik.



Gambar 2. 4 Sensor Ultrasonik
(Sumber :Rame. Is Candra, 2013)

Pada saat pengiriman ada sebuah Trigger yang digunakan sebagai pemicu agar sensor dapat mengeluarkan gelombangnya. Sementara itu Gambar 2.5 merupakan skematik dari sensor Ultrasonik yang menggambarkan rangkaian dari modul sensor.



Gambar 2. 5 Skematik Sensor Ultrasonik
(Sumber: depokinstruments.com 2016)

Tabel 2.2 merupakan fungsi dari pin sensor Ultrasonik HC-SR04. Tabel tersebut menjelaskan satu persatu fungsi dari pin sensor.

Tabel 2. 2 Fungsi Pin-pin Sensor Ultrasonik

PIN	Fungsi
VCC	Tegangan yang masuk sensor
Trig	Trigger/pemicu, digunakan untuk memicu gelombang keluar
Echo	Receive/penangkap, sebagai menangkap kembali gelombang yang terpantul pada suatu objek
GND	Sumber tegangan negatif sensor

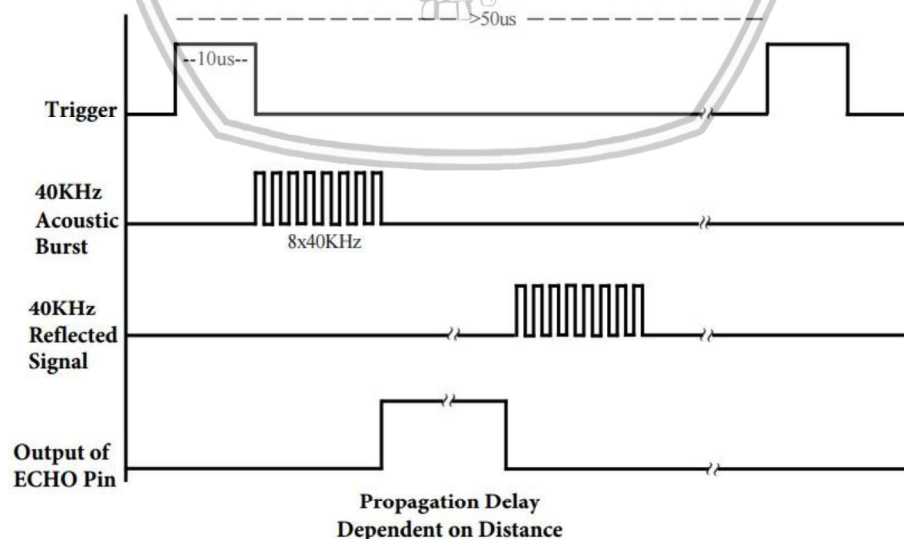
(Sumber: depokinstruments.com 2016)

a. Karakteristik HC-SR04

1. Tegangan operasi 5v
2. Arus yang diterima 15a
3. Frekuensi 40 KHz
4. Minimal mendeteksi jarak 2cm
5. Maksimal jarak 4 m
6. Sudut pantul gelombang pengukuran 15 derajat
7. Minimum waktu pemicu gelombang 10 mikrodetik

b. Diagram Waktu HC-SR04

Pada Gambar 2.6 sensor memerlukan logika 1 pada trigger untuk sebagai pemicu gelombang. Pemicu di nyalakan selama 10 mikrodetik setelah itu gelombang keluar dan terpantul sebuah objek gelombang masuk kembali ke dalam sensor dan selisih waktu saat gelombang dikeluarkan dan diterima akan dihitung dan dijadikan acuan sebagai penghitung jarak. Gelombang yang dikeluarkan sebesar 8x10KHz. (ultrasonik depokinstruments.com, 2016).

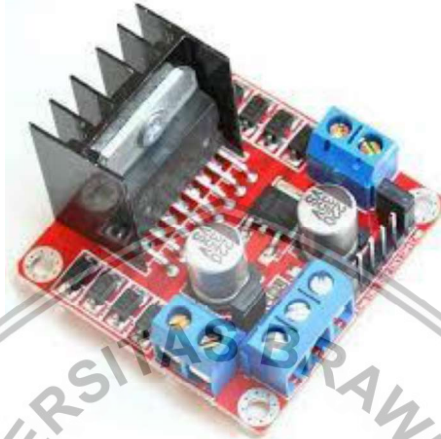


Gambar 2. 6 Diagram Clock Sensor Ultrasonik HC-SR04

(Sumber: depokinstruments.com, 2016)

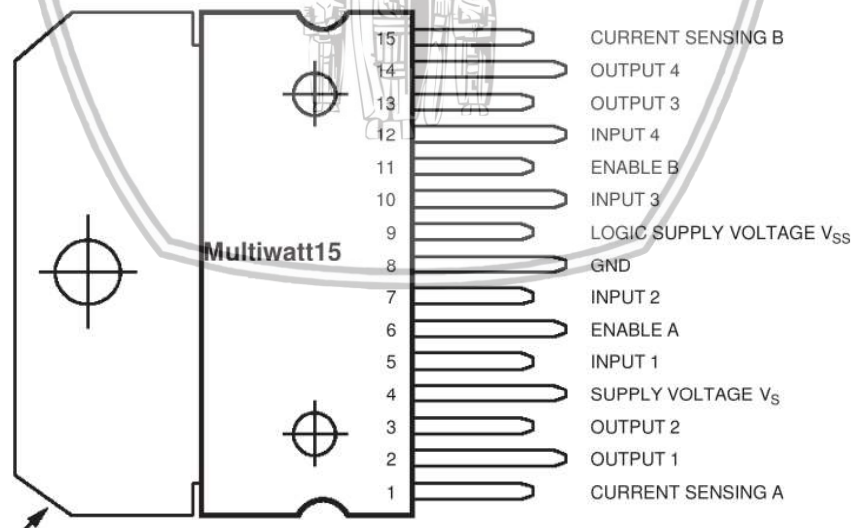
2.2.4 Driver Motor L298N

Driver motor pada Gambar 2.7 adalah suatu rangkaian yang memiliki fungsi mengatur arah dan kecepatan pada motor DC. Driver motor di desain untuk level TTL logic, mendorong beban induktif seperti relay, DC, dan stepping motors. Perangkat ini memerlukan daya mulai dari 5V hingga 46V dengan arus 2A (playground.arduino 2017).



Gambar 2. 7 Motor Driver L298N
(Sumber: playground.arduino 2017)

Gambar 2.8 merupakan konstruksi pin IC Driver Motor L298N. Terdapat 16 pin pada Driver Motor L298N ini masing-masing mempunyai fungsi yang berbeda.



Gambar 2. 8 Konstruksi Pin Driver Motor
(Suber: elektronika-dasar.web.id, 2013)

a. Fungsi Pin Driver Motor DC L298N

1. Pin EN (Enable, EN1.2, EN3.4) berfungsi untuk memungkinkan driver menerima perintah untuk menggerakan motor DC.

2. Pin In (Input, 1A, 2A, 3A, 4A) adalah pin input sinyal kendali motor DC
3. Pin Out (Output, 1Y, 2Y, 3Y, 4Y) adalah jalur output masing-masing driver yang dihubungkan ke motor DC
4. Pin VCC (VCC1, VCC2) adalah jalur input tegangan sumber driver motor DC, dimana VCC1 adalah jalur input sumber tegangan rangkaian kontrol driver dan VCC2 adalah jalur input sumber tegangan untuk motor DC yang dikendalikan.
5. Pin GND (Ground) adalah jalur yang harus dihubungkan ke ground, pin GND ini ada 4 buah yang berdekatan dan dapat dihubungkan ke sebuah pendingin kecil.

b. Fitur Driver Motor DC L298N

1. Tegangan operasi tinggi, yang bisa mencapai 40 volt;
2. Arus output besar, arus puncak sesaat dapat mencapai 3A;
3. Dengan kekuatan 25W;
4. Dua dibangun di H-jembatan, tegangan tinggi, arus besar, driver jembatan penuh, yang dapat digunakan untuk menggerakkan motor DC, motor stepper, gulungan relay dan beban induktif lainnya.
5. Menggunakan sinyal level logika standar untuk mengontrol.
6. Mampu menggerakkan motor stepper dua fase atau motor stepper empat fase, dan motor DC dua fase.
7. Mengadopsi kapasitor filter berkapasitas tinggi dan diode *freewheeling* yang melindungi perangkat di sirkuit agar tidak rusak oleh arus balik beban induktif, sehingga meningkatkan keandalan
8. Modul ini dapat menggunakan tabung *stabilivolt* tertanam 78M05 untuk mendapatkan 5v dari catu daya. Tetapi untuk melindungi chip 78M05 dari kerusakan, ketika tegangan drive lebih besar dari 12v, suplai logika 5v eksternal harus digunakan.
9. Tegangan drive: 5-35V; tegangan logika: 5V
10. Ukuran PCB: 4.2 x 4.2 cm

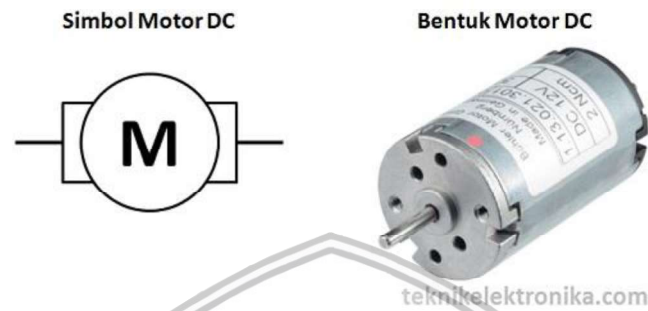
2.2.5 Motor DC

Gambar 2.9 adalah motor dc adalah suatu perangkat yang fungsinya mengubah energi listrik menjadi energi kinetik. Motor dc bekerja dengan menggunakan arus searah untuk menggerakkan motornya. Motor dc biasanya digunakan pada perangkat elektronik seperti kipas angin, bor, dan lainnya.

Motor DC berputar menghasilkan sejumlah putaran per menit dengan satuan RPM(*Revolutions per Minute*). Motor dc dapat berputar sesuai arah alur jam atau berlawanan arah alur jam jika polaritasnya dibalik. Kebanyakan motor dc dapat berputar sekitar 3000 rpm sampai 8000 rpm dengan tegangan antara 1.5v

sampai 24V. Semakin rendah tegangan yang masuk maka motor DC akan semakin melambat jika semakin tinggi maka motor DC akan berputar semakin cepat.

Jika motor DC berputar tanpa beban, daya yang dipakai sedikit, namun jika beban ditambah maka arusnya akan meningkat sampai 1000% tergantung beban yang diberikan pada motor (*Electronic Art*, 2017).



Gambar 2. 9 Simbol dan Bentuk Motor DC
(Sumber: Teknik Elektronika, 2017)

2.2.6 Power Supply

Power supply pada Gambar 2.10 merupakan sumber tegangan yang diberikan kepada seluruh komponen pada robot *maze solving*. Menggunakan baterai Lipo 12 Volt dengan tegangan +5 volt dan juga sebagai penyeimbang tegangan jika terjadi perubahan dari satu daya.

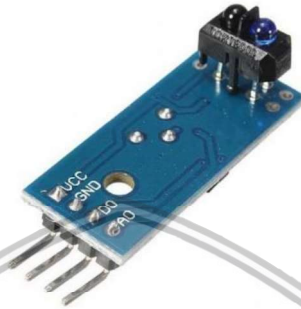


Gambar 2. 10 12 Volt Baterai
(Sumber: Manika. Anki, 2016)

Dengan adanya *Power supply*, kita tidak perlu bersusah payah mencari stop kontak atau sumber tegangan. Dengan *power supply* kit dapat menghidupkan perangkat dengan mudah dan praktis karena bisa dibawa kemana-mana. Ada dua jenis baterai yaitu bisa diisi ulang dan ada yang hanya sekali pakai (Manika. Anki, 2016).

2.2.7 Sensor Garis TCRT5000

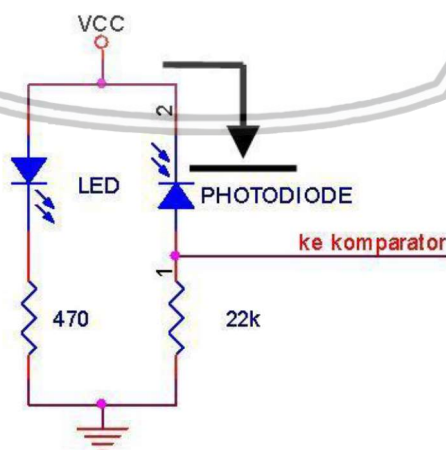
Sensor *photodiode* pada Gambar 2.11 adalah sensor yang dapat menangkap cahaya dari pantulan suatu objek. Semakin gelap objek tersebut maka daya pantul cahaya akan semakin kecil. Sensor Photodiode membentuk fungsi linear jika mengalirkan arus terhadap banyak cahaya yang diterima (Fahmi Zal, 2010).



Gambar 2. 11 Sensor Garis TCRT5000
(Sumber : ecadio.com, 2018)

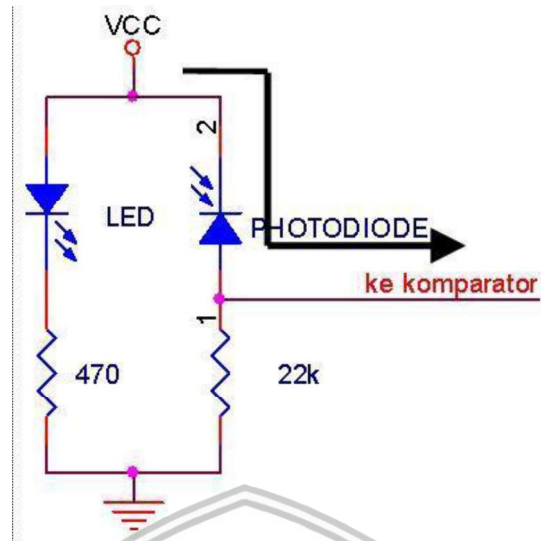
a) Prinsip Kerja Sensor

Pada Gambar 2.12 sensor photodiode dibawah ini, resistesinya berkurang jika sensor terkena intensitas cahaya dan sensor photodiode bekerja pada kondisi *reverse* bias. Untuk sumber cahaya menggunakan LED *superbright*, Led ini mempunyai cahaya cukup terang sehingga cukup untuk menyimpan intensitas cahaya ke photodiode. Gambar 2.12 sensor pada kondisi tidak terkena cahaya.



Gambar 2. 12 Sensor photodiode tidak terkena pantulan cahaya
(Sumber : Fahmi Zal, 2010)

Pada Gambar 2.13 nilai dari resistensi dari sensor akan semakin besar jika terkena cahaya.

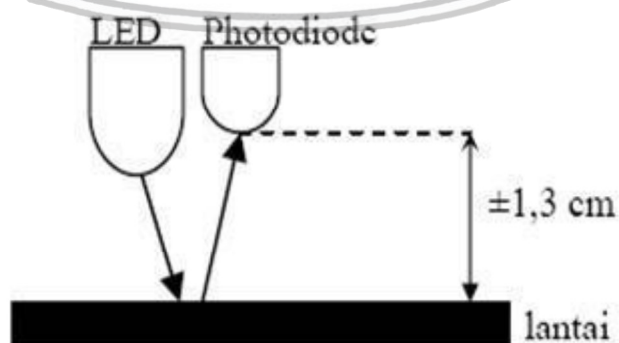


Gambar 2. 13 Sensor photodiode terkena pantulan cahaya
(Sumber : Fahmi Zal, 2010)

Gambar 2.13 terjadi saat kondisi sensor terkena pantulan cahaya. Jika terkena cahaya sumber tegangan dan resistansinya akan semakin kecil, sehingga arus yang tidak terpakai akan mengalir menuju ke komparator.

b) Mekanisme Perancangan Sensor Garis

LED berfungsi sebagai sumber cahaya yang pantulannya akan masuk ke sensor photodiode. Semakin terang permukaan yang dipantulkan maka cahaya yang masuk akan semakin besar ke dalam photodiode jika warna dari objek yang terkena pantulan semakin gelap maka cahaya yang masuk sensor photodiode akan semakin sedikit. Gambar 2.14 adalah mekanisme sensor photodiode (Fahmi Zal, 2010).

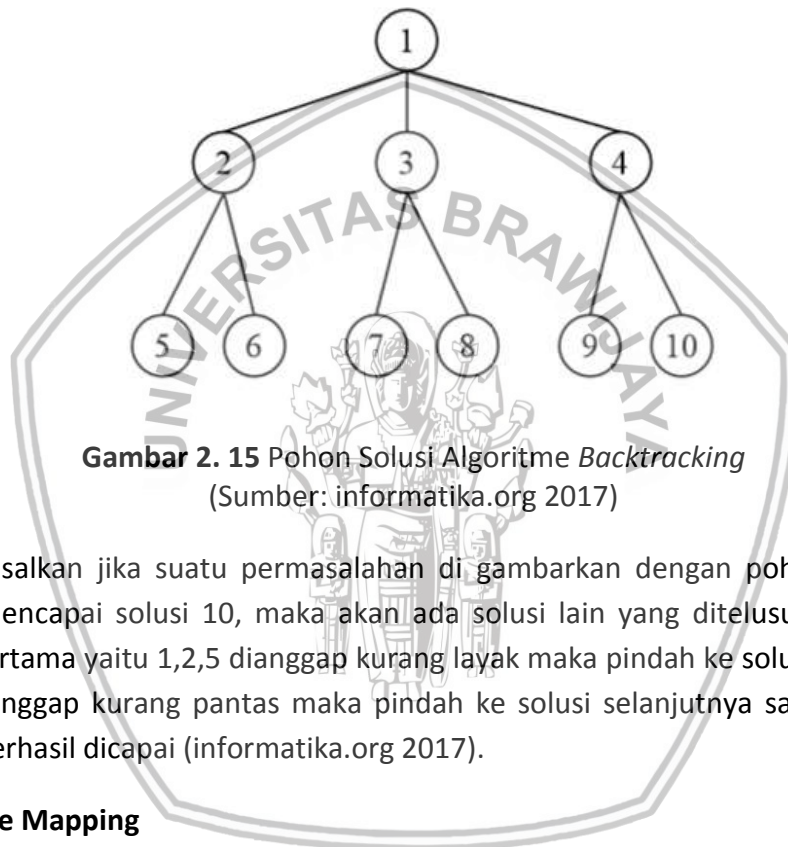


Gambar 2. 14 Ilustrasi mekanisme sensor garis
(Sumber : Fahmi Zal, 2010)

2.2.8 Algoritme Mapping dengan *Backtracking*

Pada dasarnya Algoritme *Backtracking* dan Algoritme *Brute Force* sama yaitu mencoba semua solusi dari suatu masalah. Ide dasar dari kedua Algoritme tersebut menjadi perbedaan utamanya, semua solusi yang di coba digambarkan ke dalam bentuk pohon solusi dan algoritma DFS (*depth field search*) yang akan menelusuri sampai menemukan solusi yang benar.

Algoritma *Backtrack* didapatkan dari sifat Algoritme yang memanfaatkan karakter himpunan yang solusinya sudah disusun menjadi suatu pohon solusi. Agar lebih jelas bisa dilihat pada pohon solusi pada Gambar 2.15 berikut:



Gambar 2. 15 Pohon Solusi Algoritme *Backtracking*
(Sumber: informatika.org 2017)

Dimisalkan jika suatu permasalahan di gambarkan dengan pohon di atas. Untuk mencapai solusi 10, maka akan ada solusi lain yang ditelusuri misalkan solusi pertama yaitu 1,2,5 dianggap kurang layak maka pindah ke solusi lain yaitu 1,2,6 dianggap kurang pantas maka pindah ke solusi selanjutnya sampai solusi 1,4,10 berhasil dicapai (informatika.org 2017).

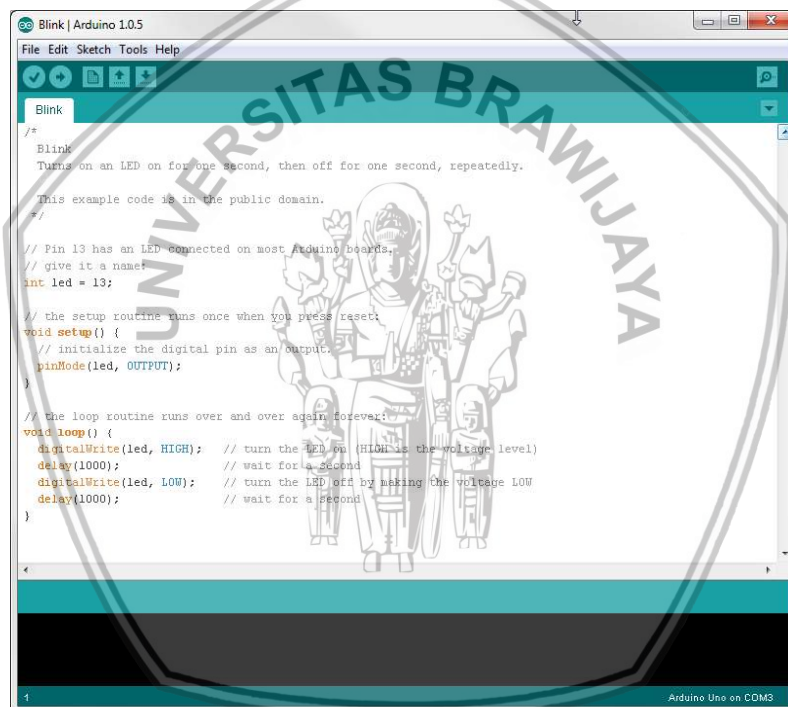
A. Maze Mapping

Maze mapping adalah suatu kegiatan robot maze follower untuk memetakan *maze* dengan cara menelusuri semua kemungkinan jalur yang akan di lewati. Maze mapping dapat menggunakan Algoritme *backtracking* karena Algoritme tersebut sangat cocok untuk memetakan sebuah *maze*. *Maze mapping* dilakukan dengan cara setiap grid akan ditelusuri dan disimpan ke dalam memori. Jadi jika robot akan menulusi kembali maka robot tidak perlu memutar *maze* lagi karena jalur yang sudah dilewati tersimpan pada memori robot (Mishra, 2008).

2.2.9 Arduino IDE

Arduino IDE adalah software bahasa pemrograman c yang digunakan pada perangkat arduino. Arduino IDE berupa *sketch* yang di dalamnya berisi library dasar yang fungsinya untuk menghubungkan perangkat lain ke Arduino. Arduino IDE memiliki 3 bagian meliputi :

1. Editor Program : untuk menulis dan mengedit program
2. Compiler : untuk mengkompilasi sebuah kode program pada *sketch*. Bisa juga tanpa mengunggah program ke Arduino. Bisa hanya sebagai pengecekan program
3. Uploader : mengunggah program hasil kompilasi ke dalam Arduino (Sumber: B.Gustomo, 2015).



Gambar 2. 16 *Sketch* Tampilan Arduino IDE

Kode program yang ada pada Arduino IDE bisa disebut *sketch* yang dibuat dengan menggunakan bahasa c. *Sketch* bisa dikelompokkan menjadi 3 blok bisa dilihat Gambar 2.16 yaitu :

1. Header : Bagian ini menulis definisi yang penting yang akan digunakan pada program selanjutnya
2. Setup : awalan dari program yang akan berjalan. Biasanya berisi konfigurasi pin output atau input pada sensor

3. Loop :bagian main program yang akan di eksekusi terus tanpa henti sampai Arduino dimatikan. Proses akan berulang ulang maka dari itu main program di namai LOOP

A. Bagian-bagian Arduino IDE

Berikut merupakan bagian-bagian dan fungsi yang ada pada *sketch* Arduino IDE yang dapat ditunjukkan pada Gambar 2.17:

1. *Verify*: merupakan istilah lain dari compile yaitu mengubah program biner untuk di upload ke Arduino
2. Upload: untuk me upload program ke dalam perangkat Arduino.
3. New Sketch: membuat sketch baru.
4. Open Sketch: Membuka kode program yang sudah pernah dibuat.
5. Save Sketch: menyimpan program
6. Serial Monitor: Membuka serial monitor secara interface.
7. Keterangan pesan-pesan yang dikerjakan oleh program Arduino IDE.
8. Konsol log: pesan yang dikerjakan. Dan pesan-pesan tentang kode program
9. Baris Sketch: untuk menunjukkan posisi baris yang sedang aktif pada kursor
10. Informasi Board dan Port: berfungsi sebagai informasi port yang di pakai oleh Arduino



Gambar 2. 17 Bagian-bagian Arduino IDE
(Sumber: Septa Ajjie, 2016)

2.2.10 EEPROM

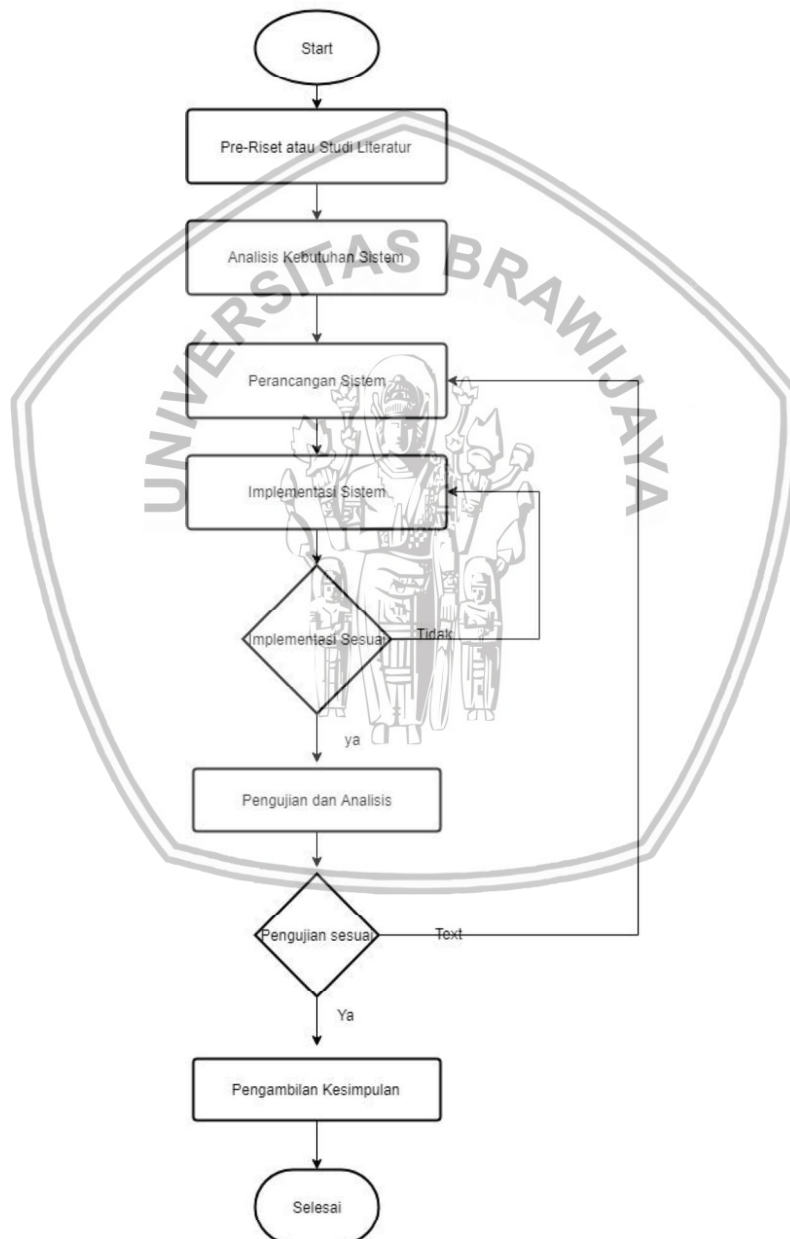
EEPROM adalah memori yang memiliki fungsi sebagai tempat media penyimpanan sebuah data jika di ibaratkan maka EEPROM adalah sebuah hardisk pada laptop atau pc, data yang masuk dan disimpan pada EEPROM tidak akan hilang meskipun catu daya di hilangkan, dimatikan atau arduino direset kecuali kita mengganti data tersebut dengan data lainnya. pada Arduino kita dapat menulis dan membaca EEPROM hingga ribuan kali. berdasarkan datasheet atmega328 dia hanya memiliki kapasitas EEPROM 1KB dan pada setiap alamat dia hanya bisa di isi dengan 8bit atau 255.

Untuk membaca dan menulis pada EEPROM pada Arduino dibutuhkan sebuah library dengan nama <EEPROM.h>. dengan library tersebut cukup hanya menuliskan data yang akan disimpan dan alamat mana yang akan menjadi tempat penyimpanan itu. Untuk library EEPROM sudah tersedia pada aplikasi Arduino IDE. Dalam library EEPROM terdapat dua fungsi utama untuk menulis data dan untuk menampilkan data yaitu :

1. EEPROM.write (address, value) : untuk menulis data ke EEPROM. Pada parameter address adalah alamat EEPROM mulai dari 0 sampai 1023, sedangkan parameter value adalah nilai atau data yang akan disimpan pada salah satu alamat pada EEPROM. Lebar data hanya 8 bit dan data dapat bernilai 0-255
2. EEPROM.read (address) : yang fungsinya untuk membaca atau mengambil data dari salah satu alamat EEPROM.

BAB III METODOLOGI PENELITIAN

Bab ini menjelaskan prinsip kerja serta, alur yang dipakai dalam peneliian ini. Alur yang digunakan adalah pre-riset atau studi literatur, analisis kebutuhan sistem yang berisi kebutuhan fungsional dan non-fungsional, perancangan sistem seperti perancangan alat dan perangkat lunak, implemantasi algoritma *backtracking*, pengujian dan analisis sistem serta cara pengambilan kesimpulan dan saran.



Gambar 3. 1 Diagram Alir Metodologi Penelitian

Berdasarkan diagram alir pada Gambar 3.1 pada tahapan di atas dilakukan secara berurutan antara studi literatur atau pre riset sampai pengambilan kesimpulan. Tahapan yang memerlukan syarat untuk melanjutkan ke tahap selanjutnya adalah implementasi sistem dan pengujian dan analisis. Pada implementasi sistem jika implementasi sesuai dengan harapan maka proses akan berlanjut ke proses pengujian dan analisis. Jika tidak sesuai maka proses itu akan diulang sampai benar-benar sesuai. Pada pengujian dan analisis jika pengujian sesuai dengan harapan maka proses akan berlanjut ke kesimpulan. Jika proses tidak sesuai maka proses akan kembali ke perancangan sistem apakah pada sistem ada yang eror atau rusak.

3.1 Pre Riset atau Studi Literatur

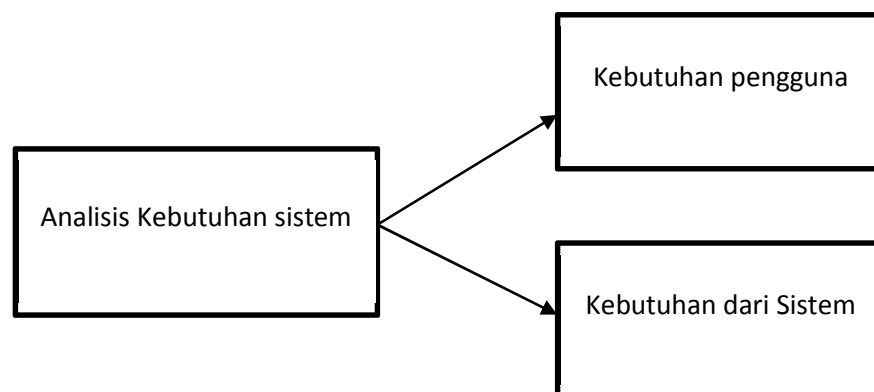
Studi literatur atau pre riset merupakan proses pengumpulan informasi yang digunakan untuk penelitian yang akan dilakukan. Prosesnya dapat berupa pengamatan lapangan, merumuskan masalah, membuat dan mengembangkan kerangka fikri, terakhir penyusunan proposal.

Tujuannya studi literatur adalah untuk memastikan permasalahan yang akan dibahas dan sebagai dasar teori dalam melakukan studi dan menjadi dasar untuk mengimplementasikan robot exploration menggunakan Algoritme mapping dengan *backtracking*. Berikut merupakan literatur yang digunakan sebagai berikut:

- 1) Teori yang mendukung mengenai *maze* robot
- 2) Mikrokontroler Arduino Nano, motor dc, sensor Ultrasonik HC SR04, sensor photodiode, L298N Driver Motor
- 3) Penerapan metode mapping dengan Algoritme *Backtracking*

3.2 Analisis Kebutuhan Sistem

Ada dua bagian dari analisis kebutuhan yaitu kebutuhan fungsional dan non fungsional.



Gambar 3. 2 Analisis Kebutuhan Sistem

Proses kebutuhan user menjelaskan tentang kebutuhan fungsional. Dari kebutuhan tersebut dapat menjelaskan beberapa poin yang dapat menjelaskan hal yang dapat user lakukan dengan sistem yang dibuat.

Pada kebutuhan fungsional pada sistem yang dirancang menjelaskan tentang kebutuhan yang diperlukan sistem agar berjalan dengan baik. Untuk kebutuhan non fungsional menjelaskan tentang batasan untuk perancangan sistem.

3.2.1 Kebutuhan Fungsional

Berikut merupakan kebutuhan fungsional yang harus ada pada sistem. Berikut merupakan kebutuhan fungsionalnya :

- 1) Mobile robot dapat mengetahui setiap node pada *wall maze*
- 2) Sistem dapat menentukan jarak sensor ultrasonik dengan setiap dinding maze
- 3) Sistem dapat menyimpan data Array ke dalam memori EEPROM
- 4) Sistem dapat menerapkan Algoritme Backtracking

3.2.2 Kebutuhan Non-Fungsional

Berikut merupakan kebutuhan non fungsional yang fungsinya untuk merancang sistem yang akan dibuat.

1) Kebutuhan Perangkat Keras:

- a. Arduino Nano
- b. Sensor ultrasonik
- c. Driver motor
- d. Sensor garis
- e. Motor dc
- f. Baterai

2) Perangkat Lunak:

- a. Arduino IDE
- b. Library NewPing.h
- c. Library EEPROM.h

3.3 Perancangan Sistem

Perancangan sistem dilakukan untuk merancang sistem yang akan dibuat adapun perancangan sistem terdiri dari perancangan perangkat keras dan perangkat lunak. Perangkat keras merancang alat-alat yang akan dirangkai untuk menjadi sistem yang utuh. Pada perancangan perangkat lunak yaitu perancangan metode dalam kode program yang akan diimplementasikan pada perangkat keras. Perancangan perangkat keras berupa sensor Ultrasonik, sensor photodiode, driver motor dc, motor dc, dan Arduino Nano. Pada perancangan perangkat lunak seperti penyimpanan data array pada memori EEPROM dan Algoritme *backtracking*.

3.4 Implementasi

Implementasi digunakan untuk merealisasikan dalam pembuatan robot *Wall Follower*. Pada implementasi perangkat keras semua perangkat terpasang dan saling terhubung. Untuk implementasi perangkat lunak robot harus sudah bisa melakukan pemetaan pada *maze* dan robot harus bisa memasukan data ke memori EEPROM.

3.5 Pengujian dan Analisis

Pengujian dilakukan pada sensor ultrasonik, sensor photodioda, sistem kendali robot dan Algoritme *backtracking*. Pengujian harus dengan data yang akurat pada saat pengujian keakuratan sensor.

3.6 Pengambilan Kesimpulan dan Saran

Pengambilan kesimpulan di dasarkan pada permasalahan yang ada dan pengujian dan analisis. Untuk saran berfungsi sebagai untuk memperbaiki kesalahan-kesalahan dan memperbaikinya pada penelitian yang akan mendatang.



BAB IV REKAYASA KEBUTUHAN

Pada bab ini menjelaskan tentang gambaran umum sistem, analisis kebutuhan sistem dan batasan desain sistem .

4.1 Gambaran Umum Sistem

Proses *mapping* pada *wall follower* pada robot *micromouse* menggunakan Algoritme *Backtracking*. *Backtracking* merupakan sebuah Algoritme sistem *mobile robot* yang dapat menentukan arah pergerakan robot pada sebuah *maze*. Dalam menentukan arah robot dapat menggunakan sensor arah *ultrasonik* dan dengan sensor arah tersebut dapat menentukan jarak antar robot dengan dinding agar tidak bertabrakan . Sensor arah *ultrasonik* robot dapat menentukan arah lurus, robot belok kanan, belok kiri, ataupun putar balik jika menemukan jalan buntu. Pada proses *mapping* robot membutuhkan input dari ketiga sensor depan, kiri dan kanan serta sensor garis untuk menentukan node pada *maze* dan dilanjutkan dengan menentukan gerakan motor dc jika belok kiri motor kanan keadaan *high* (berjalan) dan motor kiri *low* (berhenti), dan sebaliknya. Pada pengontrol kedua motor dc menggunakan driver motor agar bekerja sesuai dengan input ketiga sensor tersebut. Driver motor dapat mengotrol apakah salah satu motor mati atau kedua motor maju bersamaan. Metode dasar yang digunakan dalam proses *mapping* adalah *Right Hand Rule* yaitu setiap ada persimpangan di prioritaskan belok kanan. Pada proses *mapping* kali ini menggunakan Algoritme *Backtracking*. Algoritme *Backtracking* merupakan mencari semua kemungkinan yang ada, dengan menelusuri seluruh *maze*.

4.2 Analisis Kebutuhan Sistem

Tujuan dari kebutuhan sistem adalah agar kebutuhan yang dibutuhkan secara keseluruhan diperlukan untuk membangun *mobile robot* pada proses *mapping maze* agar sistem terpenuhi. Kebutuhan fungsional dan non fungsional sangat diperlukan untuk perancangan sistem

4.2.1 Kebutuhan Fungsional

Kebutuhan fungsional yang harus dipenuhi oleh sistem adalah :

1. Sistem dapat menentukan jarak sensor *ultrasonik* dengan setiap dinding *maze*.
 - a. Penjelasan dan Prioritas
Fungsi ini berguna agar menentukan jarak jangkauan dari sensor *ultrasonik*. Fungsi sensor tersebut adalah menentukan jarak robot dengan dinding. Dan robot akan menjaga jarak antara dinding agar tidak tertabrak.
 - b. Stimulus/respon sistem

Pada saat robot masuk maze robot akan menentukan jarak antar dinding dan menjaga jarak tetap agar tidak tertabrak di jalan lurus ataupun di persimpangan.

c. Kebutuhan Fungsional

Fungsi ini harus terpenuhi oleh sistem dikarenakan jika robot menabrak dinding akan mengganggu pergerakan robot selanjutnya

2. *Mobile robot* dapat mengetahui setiap *node* pada *Wall Maze*.

a. Penjelasan dan Prioritas

Fungsi ini berguna untuk mengetahui jarak yang sudah dilalui agar *Mobile robot* tidak dapat melewati kembali *node* yang telah dilewati. Sensor garis berfungsi sebagai mendeteksi setiap garis yang dilewati. Sistem menghitung setiap garis yang dilewati sensor agar sistem dapat mengenali *node* tersebut.

b. Stimulus/respon sistem

Pada saat sensor melewati garis *node* yang telah dipasang, sensor akan menghitung garis tersebut dengan angka agar sistem dapat mengenali garis tersebut. Dan robot tidak dapat melewati *node* yang sama.

c. Kebutuhan Fungsional

Fungsi ini harus terpenuhi oleh sistem dikarenakan Algoritme yang digunakan bertujuan menelusuri semua kemungkinan pada *maze*. Oleh karena itu robot tidak boleh melewati daerah yang sama.

3. Sistem dapat menyimpan data Array ke dalam memori EEPROM

a. Penjelasan dan Prioritas

Fungsi ini digunakan untuk menyimpan data yang telah dimasukan ke Array ke dalam memori EEPROM. Data yang dimasukkan ke dalam memori EEPROM ditujukan agar data tersebut tidak akan hilang setelah Arduino mati. Data yang masuk sebagai penanda suatu grid *maze* agar robot tidak melewati jalur yang sama

b. Stimulus/respon sistem

Data yang telah masuk ke dalam memori EEPROM tidak akan terhapus jika Arduino mati. Robot akan bergerak menuju ke grid yang belum terisi data.

c. Kebutuhan Fungsional

Fungsi ini harus terpenuhi oleh sistem dikarenakan data yang masuk ke dalam EEPROM akan di akses dan dibaca kembali oleh sistem. Oleh karena itu setiap grid *maze* harus dimasukan data agar sebagai penanda robot tidak akan melewati jalur yang sama.

4. Sistem dapat menerapkan Algoritme *Backtracking*

a. Penjelasan dan Prioritas

Dalam proses tersebut robot dapat *mapping maze* dengan mencari segala kemungkinan jalan yang ditempuh di dalam *maze*, dengan menjelajahi seluruh *maze* tanpa mengulangi jalan yang sudah ditempuh sebelumnya.

b. Stimulus/respon sistem

Degan menelusuri seluruh kemungkinan jalan yang belum ditempuh tanpa mengulangi jalan yang sama.

c. Kebutuhan Fungsional

Fungsi ini berguna untuk menampilkan output dari sistem dari proses *mapping* di dalam *maze*.

4.2.2 Kebutuhan Non-Fungsional

Berikut merupakan kebutuhan non fungsional yang terbagi menjadi 2 bagian yaitu perangkat keras dan perangkat lunak meliputi:

1. Kebutuhan Perangkat Keras

a. Arduino Nano

Sebagai pusat pemrosesan data yang dilakukan oleh mikrokontroler Atmel Atmega 168. Dan juga Arduino Nano berfungsi sebagai inti dari semua pemrosesan yang ada pada sistem

b. Sensor ultrasonik

Untuk menentukan jarak antara robot dengan dinding *maze*. Dari input sensor tersebut dapat menentukan arah dan gerakan robot.

c. Driver Motor

Mengedalikan kedua motor dc agar dapat berbelok atau maju sesuai denga input dari sensor. Driver motor tersebut dapat mengendalikan 4 motor dc sekaligus.

d. Sensor Garis

Mendeteksi garis *node* agar dapat melakukan proses mapping, dan agar robot tidak melewati jalan yang sama.

e. Motor DC

Sebagai penggerak dari robot atau sebagai roda utama dari robot.

f. Baterai

Sebagai sumber tegangan robot.

2. Kebutuhan Perangkat Lunak:

a. Arduino IDE

Arduino IDE digunakan untuk membuat program untuk prototipe sistem menggunakan Bahasa C.

b. Library NewPing.h

Digunakan untuk komunikasi antar sensor Ultrasonik dengan kontroler.

c. Library EEPROM.h

Digunakan untuk mengakses data dan menulis data pada EEPROM.

4.3 Batasan Desain Sistem

Batasan dalam pembuatan hasil jadi sistem dari robot *wall follower* yang akan dirancang. Berikut terdapat batasan desain sistem yang dibutuhkan untuk perancangan *prototype*:

1. Ukuran robot *following* yang dipakai dalam percobaan kali ini adalah 16x12 cm.
2. Sistem menggunakan 3 sensor jarak yaitu sensor ultrasonik dengan posisi depan, kanan dan kiri fungsinya sebagai pendeteksi halangan depan dan samping kanan dan kiri.
3. Untuk dapat mengendalikan 2 roda sekaligus sistem menggunakan driver motor L298N dan mikrokontroler yang dipakai adalah Arduino Nano.
4. Maze yang digunakan menggunakan ukuran 4x4 dan setiap node dibatasi garis hitam dan ukuran panjang dan lebar setiap kotak grid 30x30 cm.
5. Sistem dapat mengetahui setiap node pada *maze* dengan dibatasi garis hitam pada setiap nodenya dengan menggunakan sensor garis yang terletak bagian bawah robot.

BAB V PERANCANGAN DAN IMPLEMENTASI

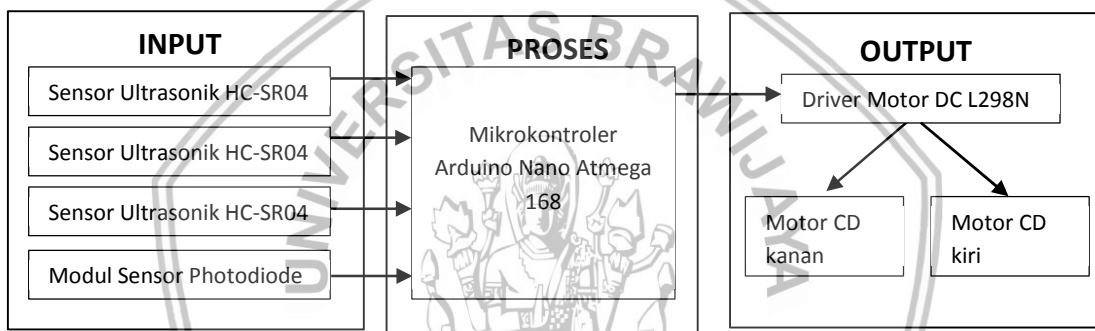
Bab ini menjelaskan mengenai perancangan prototype robot pada *Mobile robot* pada perlombaan *micromouse* dengan tipe robot *solver maze* menggunakan Algoritme *backtracking*.

5.1 Perancangan Sistem

Proses perancangan “Implementasi Algoritme *Mapping* dengan *Backtracking* pada *Mobile Robot* dengan *Maze*”. Berikut merupakan perancangan perangkat keras dan perancangan perangkat lunak.

5.1.1 Gambaran Umum Sistem

Gambar 5.1 merupakan diagram blok yang menunjukkan bagian *input*, proses dan *output*.

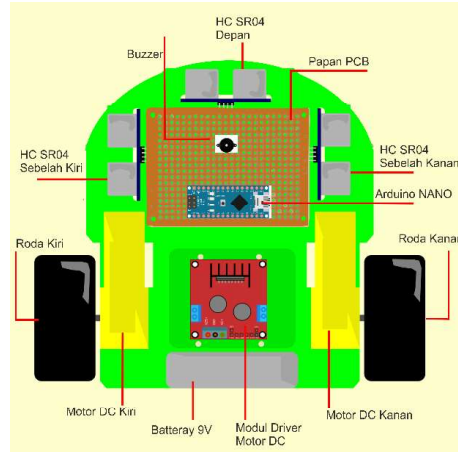


Gambar 5. 1 Diagram Blok Sistem

Pada gambaran umum diagram blok diatas merupakan gambaran yang terkait dengan rancangan perangkat dari sistem yang akan dibuat, pada bagian input terdapat 3 sensor Ultrasonik yang terletak pada bagian kanan, kiri, tengah dan 1 buah sensor photodiode pada bagian bawah. Untuk proses terdapat Arduino Nano sebagai pusat pemrosesan. Dan bagian output terdapat driver motor yang mengendalikan kedua motor.

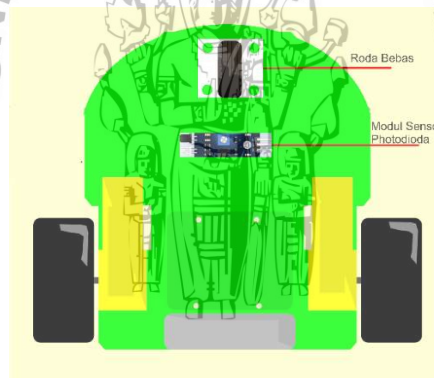
5.1.2 Perancangan *Prototype* Alat

Letak setiap komponen harus diperhatikan agar dalam tahap implementasi *prototype* dapat ditetapkan sesuai dengan yang di rencanakan pada Gambar 5.2 dan 5.3



Gambar 5. 2 Desain *Prototype* Robot Tampak Atas

Pada Gambar 5.2 adalah rancangan dari mobile robot tampak atas. Pada bagian depan terdapat tiga sensor ultrasonik yang terletak pada depan, kanan dan kiri. Di sebelah tengah terdapat driver motor yang terpasang pada pcb dan samping pcb terdapat Arduino Nano yang terpasang pada bagian tengah belakang. Pada samping Arduino Nano terlihat roda robot tampak dari atas.



Gambar 5. 3 Desain *Prototype* Robot Tampak Bawah

Pada Gambar 5.3 adalah robot tampak bagian bawah terdapat bagian depan roda 360 derajat yang bisa berputar 360 derajat. Samping roda 360 derajat terdapat sensor photodiode yang menghadap ke bawah sebelah belakang terdapat roda dan motor dc.

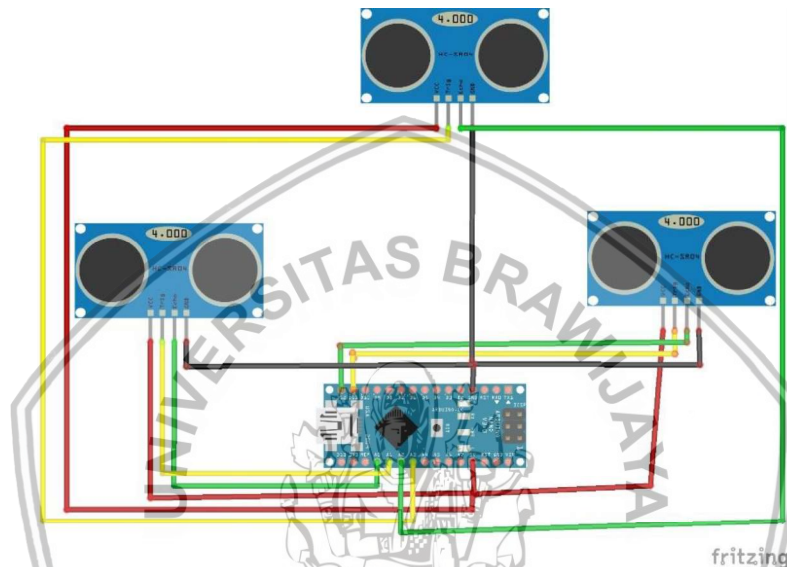
5.1.3 Perancangan Perangkat Keras

Pada blok diagram pada Gambar 5.1 pada perancangan perangkat keras . pada bagian utamanya terdapat Arduino Nano yang menjadi pusat pemrosesan datanya. Pada bagian input terdapat 3 sensor Ultrasonik yang terletak pada sebelah depan, kanan, dan kiri serta photodiode yang terletak pada bagian

bawah. bagian output terdapat driver motor yang mengendalikan kedua motor dc tersebut.

5.1.3.1 Perancangan Sensor Ultrasonik

Gambar 5.4 merupakan gambar konfigurasi pin dari ketiga sensor Ultrasonik. Pada perancangan sensor Ultrasonik terdapat 3 sensor Ultrasonik yang terletak pada kanan kiri dan depan robot. Untuk Penjelasan dari konfigurasi pin terdapat pada Tabel 5.1.



Gambar 5. 4 Konfigurasi Pin Sensor Ultrasonik

Tabel 5.1 Merupakan keterangan dari konfigurasi pin yang ada pada Gambar 5.4. Pada tabel ini menunjukkan letak pemasangan pin dari sensor Ultrasonik.

Tabel 5. 1 Konfiguasi Pin Sensor Ultrasonik

Sensor kiri	Jarak	Sensor kanan	Jarak	Sensor Tengah	Jarak	Arduino NANO
VCC						5v
Trigger						A0
Echo						A1
GND						GND
		VCC				5v
		Trigger				11
		Echo				12
		GND				GND
				VCC		5v
				Trigger		A2
				Echo		A3
				GND		GND

Tabel 5.1 menunjukkan konfigurasi pin Antara Arduino NANO dan perangkat keras lainnya secara terperinci. Pada ketiga sensor Ultrasonik masing-masing pin trigger kanan terpasang pada pin Digital 11 Arduino NANO, trigger kiri terpasang pada pin analog A0 Arduino NANO, trigger depa terpasang pada pin analog 2 Arduino NANO. Serta VCC dan Ground terpasang pada pin 5v dan Ground pada Arduino Nano.

5.1.3.2 Perancangan Driver Motor DC L298N

Gambar 5.2 merupakan konfigurasi dari pin Driver Motor DC L298N. Terdapat 2 motor yang dikendalikan oleh driver motor. Fungsi dari driver motor adalah untuk mengendalikan arah putaran dan kecepatan dari kedua motor tersebut.



Gambar 5. 5 Konfigurasi Pin Driver Motor DC L298N

Tabel 5.2 Merupakan keterangan dari konfigurasi pin yang ada pada Gambar 5.5. Pada tabel ini menunjukkan letak pemasangan pin dari driver motor.

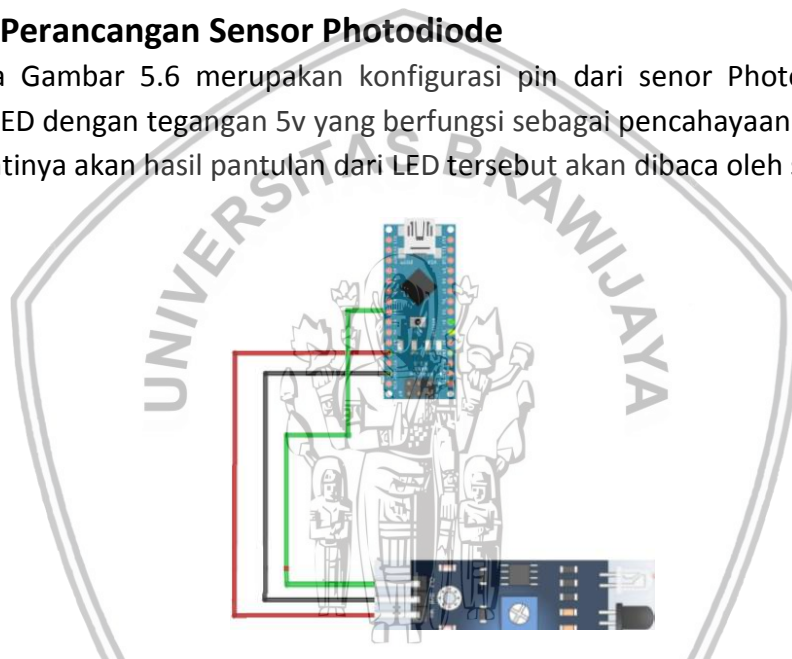
Tabel 5. 2 Konfigurasi Pin Driver Motor DC L298N

Motor Kiri	Motor Kanan	Driver Motor L298N	Arduino Nano
Out1		Out1	
Out2		Out2	
	Out1	Out3	
	Out2	Out4	
		Enable 1	D3
		Enable 2	D9
		Input 1	D4
		Input 2	D5
		Input 3	D7
		Input 4	D6
		Vin	Vin
		VCC	5v
		Gnd	Gnd

Pada Tabel 5.2 2 Motor DC terpasang langsung pada IC Driver Motor L298N pada motor kiri Out1 terpasang pada pin Out 1 pada Driver Motor, Out2 terpasang pada pin Out 2 pada Driver Motor, pada motor kanan Out1 terpasang pada pin Out 3 pada Driver Motor, dan Out2 terpasang pada pin Out 4 pada Driver Motor. Pin driver motor yang terhubung pada pin Arduino Nano adalah Enable1 terpasang pada pin D3 digital, enable2 terpasang pada pin D9 digital, input1 terpasang pada D4 digital, input2 terpasang pada D5 diital, input3 terpasang pada pin D7 digital dan pin input4 terpasang pada pin D6 digital Arduino Nano. Serta VCC dan Ground terpasang pada pin 5v dan Ground pada Arduino Nano.

5.1.3.3 Perancangan Sensor Photodiode

Pada Gambar 5.6 merupakan konfigurasi pin dari senor Photodiode dan sebuah LED dengan tegangan 5v yang berfungsi sebagai pencahayaan dari sensor yang nantinya akan hasil pantulan dari LED tersebut akan dibaca oleh sensor.



Gambar 5. 6 Konfigurasi Pin Sensor Photodiode

Tabel 5.3 Merupakan keterangan dari konfigurasi pin yang ada pada Gambar 5.6. Pada tabel ini menunjukkan letak pemasangan pin dari Sensor Photodiode.

Tabel 5. 3 Konfigurasi Pin Sensor Photodiode

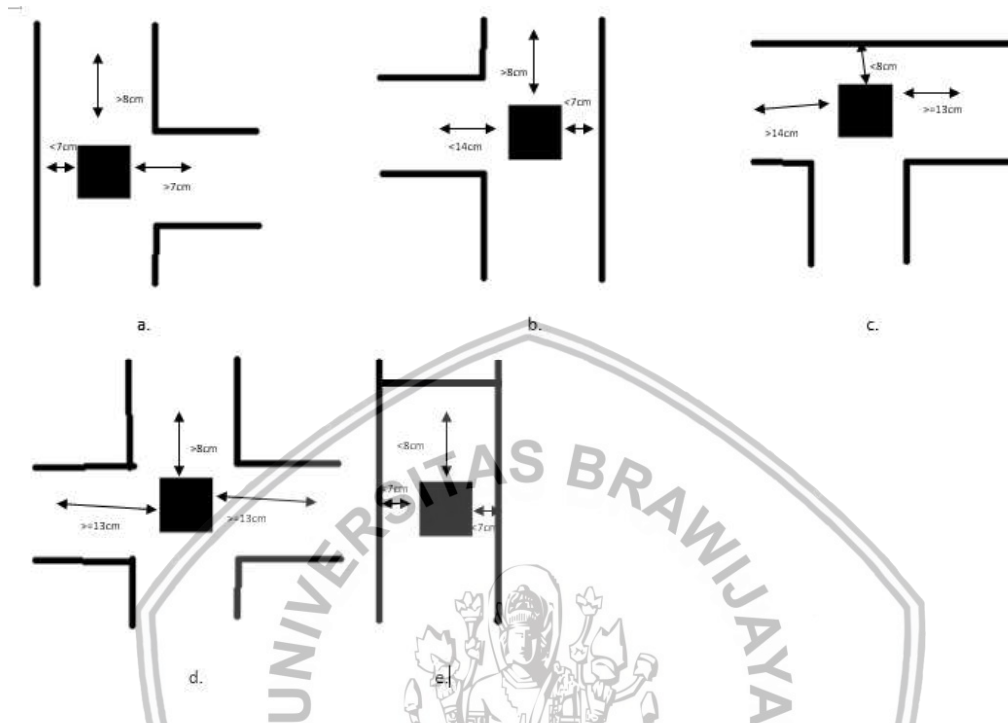
Sensor Photodiode	Arduino Nano
VCC	5v
Gnd	Gnd
Out	10

5.1.4 Perancangan Perangkat Lunak

Pada sub bab ini menjelaskan perancangan dari perangkat lunak. Mulai dari perancangan *maze* , Array 2 Dimensi dan *flowchart*.

5.1.4.1 Perancangan Pembacaan Sensor Ultrasonik HC-SR04

Dalam *flowchart* pada Gambar 5.7 data yang digunakan adalah jarak antara objek dinding dengan sensor.

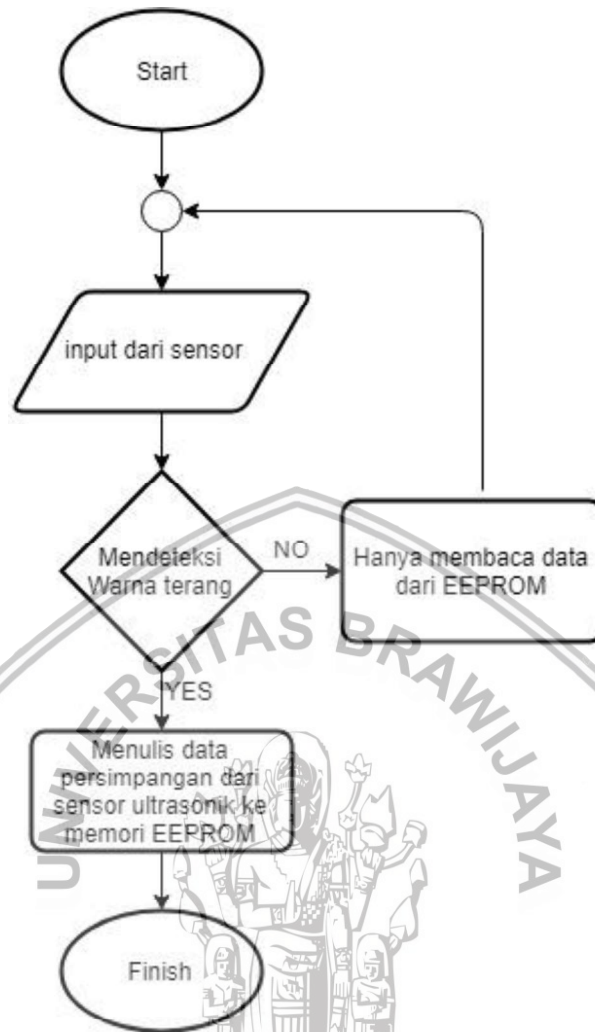


Gambar 5. 7 Pembacaan Sensor Ultrasonik

Sensor Ultrasonik diletakkan di bagian kiri, depan dan kanan robot. Sensor Ultrasonik dihadapkan menuju dinding. Pada Gambar 5.7 tersebut jika jarak sensor ke dinding kiri kurang dari sama dengan 7 cm, depan lebih dari 8 dan kanan lebih dari 7cm maka robot akan belok kanan (a). Jika kiri lebih dari 14cm, kana kurang dari 7cm dan depan lebih dari 8cm maka jalan lurus (b). Jika kiri lebih dari 14zm, kanan lebi dari 13cm dan depan kurag dari 9cm maka akan belok kanan (c). Jika kiri lebih dari 13 kanan lebih dari 8 dan depan lebih dari samadengan 13 maka akan belok kanan (d). Jika kiri kurang dari sama dengann 7 kanan kurang dari sama dengan 7 dan depan kurang dari sama dengann 8 maka akan putar balik (e).

5.1.4.2 Perancangan Pembacaan Modul Sensor Photodiode

Pada Gambar 5.8 data yang diambil adalah nilai tegangan dari perhitungan dari resistensi sensor. Jika sensor mmendeteksi warna terang maka resistansi dari sensor semakin besar maka dari itu sensor mendeteksi garis hitam.

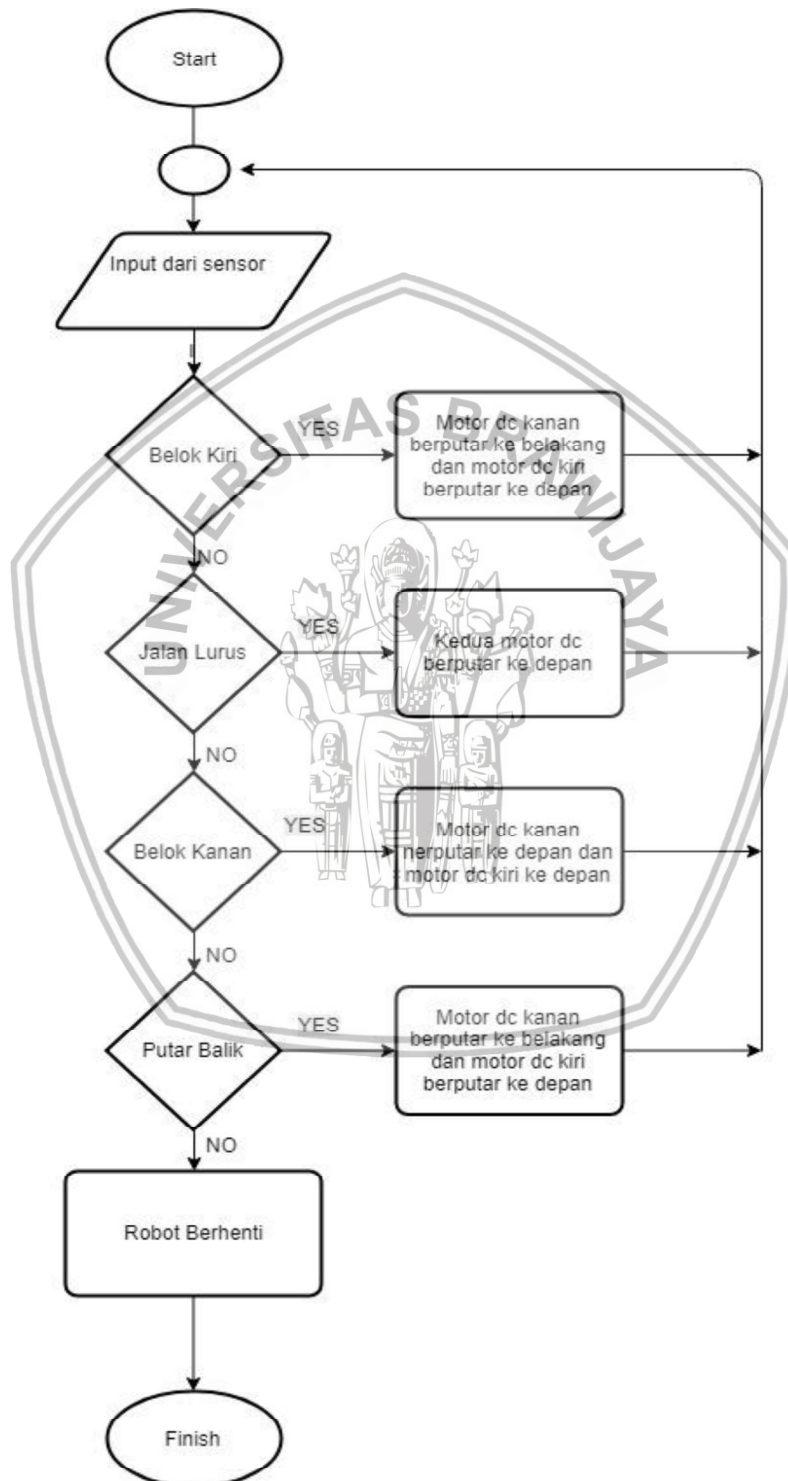


Gambar 5. 8 Flowchart Sensor Photodiode

Modul sensor Photodiode yang digunakan satu buah. Sensor diletakkan di bagian bawah robot dengan sensor menghadap ke arah bawah menuju dasar *maze*. Jarak antar sensor dengan dasar labirin lebih kurang 2.5 cm. Menggunakan pin 6 digital Arduino Nano dengan fungsi PWM di dalamnya. Pada setiap batas *Grid* pada labirin telah ditandai dengan warna gelap. Saat melewati batas *Grid* yang berwarna gelap maka bernilai LOW atau 0 dan bagian terang bernilai HIGH atau 1. Robot akan mengenali setiap *Grid* jika sensor melewati semua batas *Grid* tersebut. Jika robot bernilai mendeteksi warna gelap maka robot akan membaca persimpangan jarak antara ketiga sensor dan dinding apakah robot itu belok kanan di simbolkan angka 1 atau lurus disimbolkan angka 0 atau belok kiri disimbolkan angka 2 dan putar balik disimbolkan angka 3

5.1.4.3 Perancangan Menentukan Putaran Motor DC pada L298N Driver Motor

Pada Gambar 5.9 data yang digunakan adalah dari putaran motor dan arah putar motor yang dikendalikan oleh driver motor dc L298N.

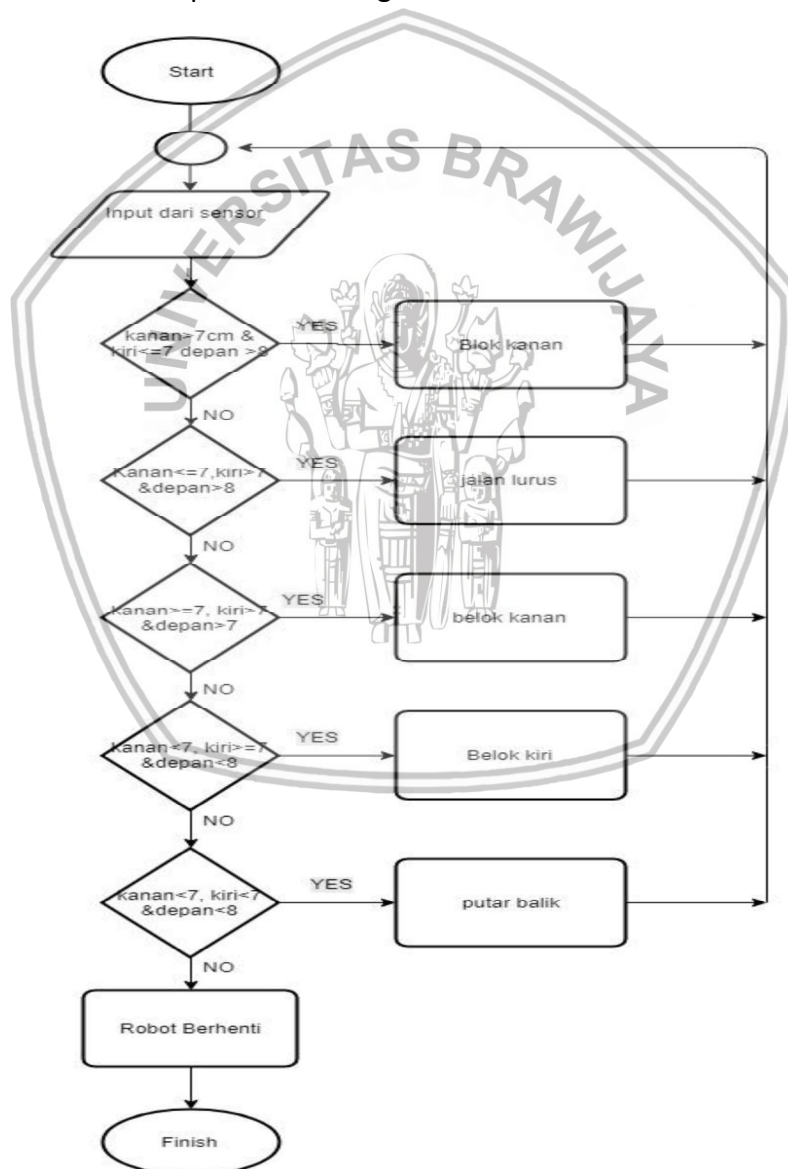


Gambar 5. 9 Flowchart L298N Driver Motor

Penggunaan iC driver motor menggunakan 1 buah yang dapat mengendalikan kedua motor. Jika belok kiri yang berputar depan adalah motor dc kanan dan yang kanan berputar ke belakang. Jika lurus maka kedua motor akan bergerak ke depan. Jika belok kiri roda kanan berputar ke depan dan roda kiri berputar ke belakang. Jika putar balik sama denga belok kanan akan tetapi durasi dari perputarannya lebih lama.

5.1.4.4 Perancangan Pergerakan Robot *Follow Right*

Berdasarkan pada gambar 5.7 pada Gmabar 5.10 inputan yang digunakan dari jarak yang dibaca sensor Ultrasonik. Dari waktu pantulan gelombang yang dikeluarkan sensor sampai diterima lagi sensor.

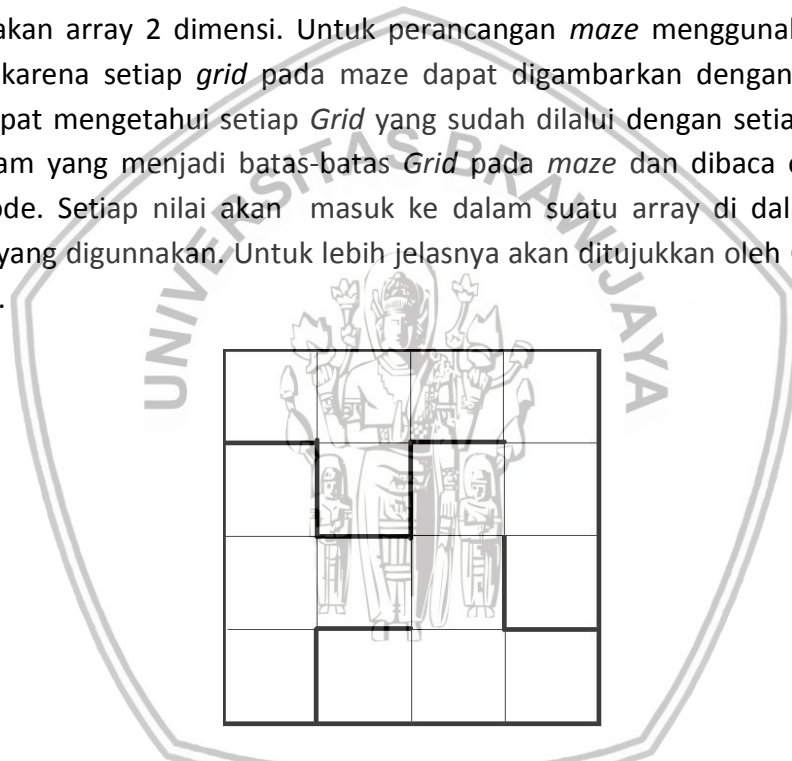


Gambar 5. 10 *Follow Right*

Pergerakan *follow right* yang dapat dijelaskan dengan *flowchart* pada Gambar 5.10 di atas. Pergerakan *follow right* adalah pergerakan dasar dari robot *maze solver* ini. Pada pergerakannya memprioritaskan untuk berbelok ke kanan dari pada berbelok ke kiri ataupun jalan lurus ke depan. Jika sensor ultrasonik mendeteksi kiri kurang dari 7cm kanan lebih dari 7cm dan depan kurang lebih dari 8cm maka robot akan belok kanan. Jika kiri kurang 7cm kanan kurang 7cm dan depan lebih 8cm maka robot akan jalan lurus. Jika kiri lebih 8cm kanan kurang 7cm dan depan kurang 7cm maka robot akan jalan lurus.

5.1.4.5 Perancangan *Maze* Menggunakan Array 2 Dimensi

Pada poin ini akan menjelaskan tentang perancangan *maze* yang menggunakan array 2 dimensi. Untuk perancangan *maze* menggunakan array 2 dimensi karena setiap *grid* pada *maze* dapat digambarkan dengan array. Jika robot dapat mengetahui setiap *Grid* yang sudah dilalui dengan setiap melewati garis hitam yang menjadi batas-batas *Grid* pada *maze* dan dibaca oleh sensor Photodiode. Setiap nilai akan masuk ke dalam suatu array di dalam array 2 dimensi yang digunakan. Untuk lebih jelasnya akan ditunjukkan oleh Gambar 5.1 dan 5.13.



Gambar 5. 11 Rancangan *maze* yang digunakan

Pada Gambar 5.11 merupakan tampilan *maze* yang di simulasikan ke dalam Array 4x4 dengan ukuran per node 30x30 cm dalam *maze* tersebut setiap *grid* terdapat batas berupa garis hitam sebagai penanda sensor untuk memberi input ke dalam array. Pada Gambar 5.11 merupakan gambar *maze* yang disimulasikan ke dalam array dan setiap *grid* di beri variabel agar memudahkan untuk mengakses setiap *grid* pada array tersebut.

a	b	c	d
e	f	g	h
i	j	k	l
m	n	o	p

Gambar 5. 12 Variabel pada Setiap Array

Pada setiap array dinamai setiap variabel yang digunakan contoh $maze[0][0] = a$, $maze[0][1] = b$ dan seterusnya. Pemberian nama pada setiap array akan memudahkan untuk mengakses array tersebut pada saat disimpan pada EEPROM. Nilai dari sensor photodiode yang di dapat dari melewati *Grid* yang berwarna gelap akan terisi pada variabel yang telah tersimpan pada EEPROM. Variabel sudah terisi array tersebut akan dinyatakan sudah dilewati oleh robot. Jika robot melewati variabel atau *Grid* yang sudah tersisi maka nilai dari sensor Photodiode akan diabaikan dan robot akan terus sampai menemukan *Grid* yang belum terisi. Jika variabel di dalam array tersebut sudah terisi semua maka robot akan otomatis berhenti.

5.1.4.6 Penyimpanan ke Memori EEPROM Arduino Nano

Penyimpanan pada EEPROM digunakan untuk meyimpan array yang nantinya akan di akses sebagai tempat penyimpanan nilai *Grid* yang didapatkan dari sensor photodiode. Saat sensor melewati garis hitam sebagai batas suatu node nilai sensor akan dimasukkan ke satu *Grid* array sebagai tanda robot telah melewati *Grid* tersebut.

Pada Gambar 5.12 menjelaskan bahwa setiap *Grid* diberi variabel huruf seperti pada gambar. Variabel pada array tersebut tersimpan pada EEPROM agar tidak mudah terhapus. Jika akan mengisi variabel tersebut dengan angka maka harus mengakses variabel di dalam EEPROM terlebih dahulu agar data yang telah diisi tidak akan terhapus jika robot dimatikan.

5.1.4.7 Maze Mapping Robot dengan Algoritme Backtracking

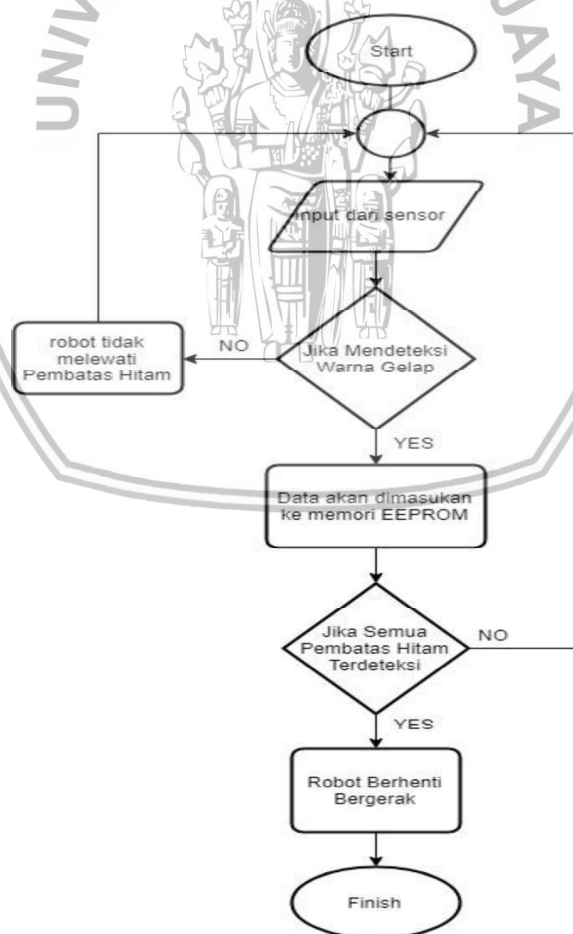
Pada Algoritme *Maze Mapping* robot akan memetakan seluruh area pada maze. Memetakan maze dengan memberi tanda pada setiap batas-batas *Grid*. Setiap kali robot melewati batas tersebut sensor Photodiode akan mendeteksi dan memberi nilai pada *Grid* tersebut sampai semua *Grid* yang dilewati diberi nilai. Jika seluruh *Grid* seperti pada Gambar 5.13 telah berisi nilai maka robot otomatis akan berhenti bergerak. Robot tidak akan memberi nilai pada *Grid* yang

sudah dilewati. Jadi robot akan memberi nilai hanya satu kali pada *Grid* yang kosong.

1	2	3	4
10	3	6	5
9	8	7	6
10	9	8	9

Gambar 5. 13 Penomoran Grid pada Array

Pada Gambar 5.14 jika sensor melewati garis hitam maka sensor akan menginput data ke memori. Jika sensor melewati dasar yang terang maka sensor tidak akan melakukan input ke memori.



Gambar 5. 14 Flowchart Algoritme Mapping dengan Input Sensor Photodiode

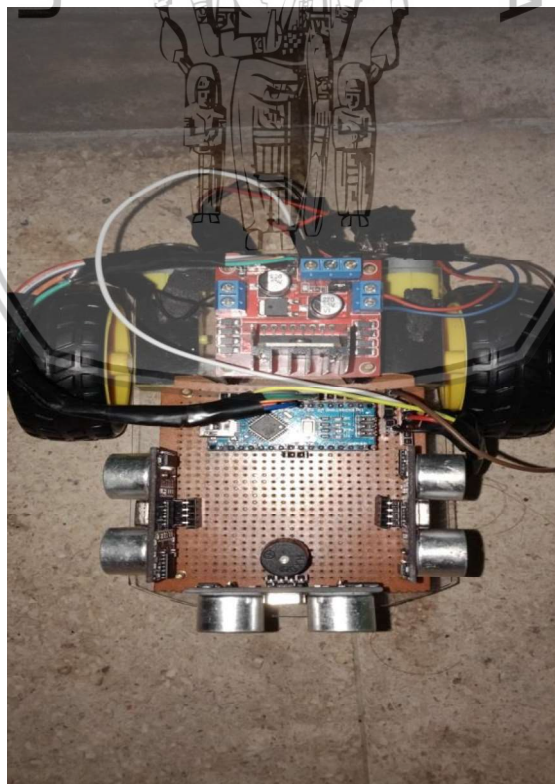
5.2 Implementasi

Pada tahap implementasi adalah tahap dimana dari perancangan rancang agar robot dapat direalisasikan.

5.2.1 Implementasi *Prototype* Alat

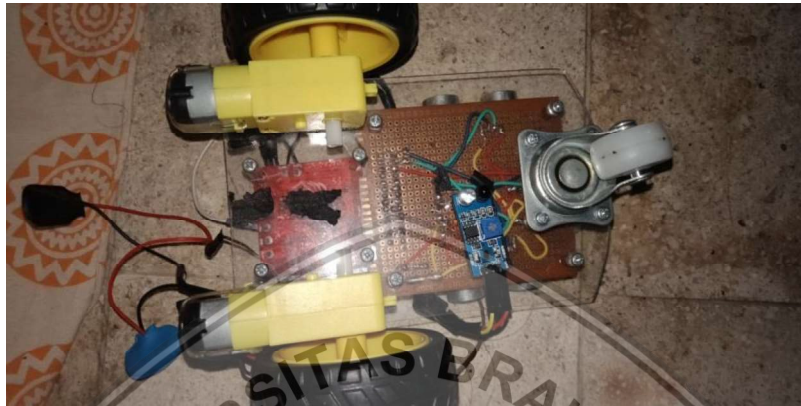
Pada Gambar 5.15 di bawah ini ukuran dari robot adalah 14x 12 cm. Untuk bahan penjangnya agar semua komponen melekat adalah acrylic yang menjadi landasan dari semua komponen. Pada sisi tengah robot terdapat pcb tepat driver motor melekat dan untuk tempat baterainya terletak pada bagian bawah robot. Untuk karton yang digunakan adalah karton yang tebal dan dapat menampung dengan kuat semua komponen yang melekat padanya.

Pada Algoritme *Maze Mapping* robot akan memetakan seluruh area pada maze. Memetakan maze dengan memberi tanda pada setiap batas-batas *Grid*. Setiap kali robot melewati batas tersebut sensor Photodiode akan mendeteksi dan memberi nilai pada *Grid* tersebut sampai semua *Grid* yang dilewati diberi nilai. Jika seluruh *Grid* seperti pada Gambar 5.13 telah berisi nilai maka robot otomatis akan berhenti bergerak. Robot tidak akan memberi nilai pada *Grid* yang sudah dilewati. Jadi robot akan memberi nilai hanya satu kali pada *Grid* yang kosong.



Gambar 5. 15 Robot Bagian Atas

Yang melekat pada papan *acrylic* sebelah depan adalah 3 sensor ultrasonik yang ditempatkan di depan kanan dan kiri agar dapat mengetahui objek dinding disekitarnya. Arduino menempel pada bagian belakang *acrylic* dan disebelahnya terdapat roda yang terlihat dari atas robot. Untuk driver motor nya tidak langsung melekat pada *acrylic* akan tetapi driver motornya yang berada pada tengah robot.

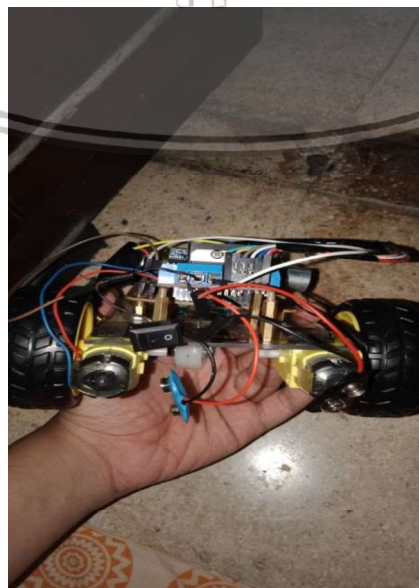


Gambar 5. 16 Robot Bagian Bawah

Pada bagian bawah terdapat dua motor dc kit yang terpasang pada bagian belakang karton. Untuk sensor photodiodanya terletak pada sebelah roda 360 derajat.

5.2.2 Implementasi Perangkat Keras

Implementasi perangkat keras dapat dilihat pada Gambar 5.17 merupakan robot bagian belakang. Komponennya adalah Arduino Nano, Sensor Ultrasonik 3 buah sensor Photodiode 1 buah dan 2 motor dc yang dikendalikan oleh Icdriver motor L298N



Gambar 5. 17 Robot Bagian Belakang

Pada gambar dibawah L298N terpasang pada papan PCB dan kabel jumper yang terpasang pada pin yang tersambung pada Arduino dan 2 motor DC. 3 sensor Ultrasonik terpasang pada sisi kanan, kiri dan depan robot. Sensor Photodiode terpasang pada bagian bawah robot dan menghadap kebawah serta baterai 9v diletakkan pada bagian atas Arduino Nano.



Gambar 5. 18 Robot Bagian Belakang

5.2.3 Implementasi Perangkat Lunak

Implementasi perangkat lunak menjelaskan program yang ditulis pada sketch arduino. Perancangan perangkat lunak dilakukan agar dapat memfungsikan semua komponen dengan baik dan merealisasikan Algoritma dengan baik.

5.2.3.1 Implementasi Pembacaan Sensor Ultrasonik

Di bawah ini merupakan implementasi perangkat lunak dari sensor ultrasonik. Pada program di bawah ini bermaksud untuk menghitung jarak antar sensor dengan dinding.

Tabel 5. 4 Pembacaan Sensor Ultrasonik

No	Source Code
1	void loop()
2	{
3	/* input dan output sinyal masing-masing
4	dideklarasikan
5	Sensor ultrasonik dan disimpan oleh variabel sensor
6	yang mengubah kecepatan suara yang 340 m / s atau
7	29 mikrodetik per sentimeter, saat sinyal berjalan
8	bolak balik
9	kali ini sensor = waktu / 29/2;
	begitu juga pada dua sensor lainnya. */

10	<code>digitalWrite(trigger_depan, LOW);</code>
11	<code>delayMicroseconds(2);</code>
12	<code>digitalWrite(trigger_depan, HIGH);</code>
13	<code>delayMicroseconds(5);</code>
14	<code>digitalWrite(trigger_depan, LOW);</code>
15	<code>durasi_depan = pulseIn(echo_depan, HIGH);</code>
16	<code>depan = durasi_depan/29/2;</code>
17	<code>digitalWrite(trigger_kiri, LOW);</code>
18	<code>delayMicroseconds(2);</code>
19	<code>digitalWrite(trigger_kiri, HIGH);</code>
20	<code>delayMicroseconds(5);</code>
21	<code>digitalWrite(trigger_kiri, LOW);</code>
22	<code>durasi_kiri = pulseIn(echo_kiri, HIGH);</code>
23	<code>kiri = durasi_kiri/29/2;</code>
24	<code>digitalWrite(trigger_kanan, LOW);</code>
25	<code>delayMicroseconds(2);</code>
26	<code>digitalWrite(trigger_kanan, HIGH);</code>
27	<code>delayMicroseconds(5);</code>
28	<code>digitalWrite(trigger_kanan, LOW);</code>
29	<code>durasi_kanan = pulseIn(echo_kanan, HIGH);</code>
30	<code>kanan = durasi_kanan/29/2;</code>
31	

Pada baris 3 sampai 8 menjelaskan tentang input dan output sinyal masing-masing dideklarasikan sensor Ultrasonik dan disimpan oleh variabel sensor yang mengubah kecepatan suara sebesar 340 m/s atau 29 mikrodetik per sentimeter saat sinyal berjalan bolak balik. Maksud dari baris 8 adalah membagi waktu sensor yang mulanya 29 mikrodetik dibagi dengan 2. Hal yang sama dilakukan kedua sensor lainnya.

5.2.3.2 Implementasi Pembacaan Modul Sensor Photodiode

Tabel 5. 5 Pembacaan Modul Sensor Photodiode

No	Source Code
1	<code>void setup() {</code>
2	<code>Serial.begin(9600);</code>
3	<code>pinMode(6,INPUT); // Open serial monitor at 9600 baud</code>
4	<code>to see ping results.</code>
5	<code>}</code>
6	<code>void loop() {</code>
7	<code>delay(500);</code>
8	<code>int sensor=6;</code>
9	
10	<code>if (sensor==LOW)</code>
11	<code>{</code>
12	<code>Serial.println("Sensor deteksi: hitam(on)");</code>
13	<code>Serial.println(" ");</code>
14	<code>}</code>
15	<code>if (sensor==HIGH)</code>

16	{
17	Serial.println("Sensor deteksi: putih(off)");
18	Serial.println(" ");
19	}

Pada pembacaan sensor Photodiode pada source code terdapat 2 keadaan yaitu *HIGH* dan *LOW*. *HIGH* pembacaan dasar *maze* dengan warna cerah sedangkan *LOW* melakukan pembacaan batas-batas *Grid* yang berwarna gelap.

5.2.3.3 Implementasi Driver Motor L298N

Tabel 5. 6 Pembacaan IC Driver Motor L298N

No	Source Code
1	void left() // berfungsi untuk membuat robot
2	berputar 90° ke kiri jika tidak memiliki output ke depan
3	dan kanan;
4	{
5	analogWrite(kecepatan_motor_kiri, 120);
6	analogWrite(kecepatan_motor_kanan, 120);
7	
8	analogWrite(e1, 0);
9	analogWrite(e2, 255);
10	analogWrite(d1, 0);
11	analogWrite(d2, 255);
12	delay(700);
13	}
14	void right() // berfungsi untuk membuat robot
15	berputar 90° ke kanan jika tidak memiliki output depan
16	atau kiri;
17	{
18	
19	analogWrite(kecepatan_motor_kiri, 120);
20	analogWrite(kecepatan_motor_kanan, 120);
21	
22	analogWrite(e1, 255);
23	analogWrite(e2, 0);
24	analogWrite(d1, 255);
25	analogWrite(d2, 0);
26	delay(800);
27	}
28	
29	void back() // berfungsi menyebabkan robot berputar
30	180° jika tidak ada output depan, kanan dan kiri;
31	{
32	
33	analogWrite(kecepatan_motor_kiri, 120);
34	analogWrite(kecepatan_motor_kanan, 120);
35	
36	analogWrite(e1, 255);
	analogWrite(e2, 0);

37	<code>analogWrite(d1, 255);</code>
38	<code>analogWrite(d2, 0);</code>
39	<code>delay(1200);</code>
40	
41	<code>}</code>

Pada sourcecode di atas variabel e1 berarti mengendalikan arah rotasi motor kiri ke depan, e2 berarti mengontrol arah rotasi motor kiri ke belakang, d1 berarti mengontrol arah rotasi motor kanan ke belakang, d2 mengontrol arah rotasi motor kanan ke depan. Nilai PWM terbesar dari motor DC adalah 255. Pada void left e2 bernilai 255 dan d2 bernilai 255 yang berarti motor kiri berputar ke belakang sedangkan yang kanan berputar ke depan. Pada void right e1 bernilai 255 dan d1 bernilai 255 yang berarti motor kiri berputar ke depan sedangkan yang kanan berputar ke belakang. Pada void back e1 bernilai 255 dan d1 bernilai 255 yang berarti motor kiri berputar ke depan sedangkan yang kanan berputar ke belakang.

5.2.3.4 Implementasi Pergerakan *Follow Right*

Tabel 5. 7 Pergerakan *Follow Right*

No	Source Code
1	<code>if (depan >8)</code>
2	<code>{</code>
3	<code> if (kanan >7 && kanan< 13)</code>
4	<code> {</code>
5	<code> analogWrite(kecepatan_motor_kiri, 200);</code>
6	<code> analogWrite(kecepatan_motor_kanan, 200);</code>
7	
8	<code> analogWrite(e1, 255);</code>
9	<code> analogWrite(e2, 0);</code>
10	<code> analogWrite(d1, 0);</code>
11	<code> analogWrite(d2, 255);</code>
12	<code> }</code>
13	
14	<code> if (kanan >=13)</code>
15	<code> {</code>
16	<code> analogWrite(kecepatan_motor_kiri, 255);</code>
17	<code> analogWrite(kecepatan_motor_kanan, 60);</code>
18	
19	<code> analogWrite(e1, 255);</code>
20	<code> analogWrite(e2, 0);</code>
21	<code> analogWrite(d1, 0);</code>
22	<code> analogWrite(d2, 255);</code>
23	<code> }</code>
24	
25	
26	<code> if (kanan <=7)</code>
27	<code> {</code>
28	<code> analogWrite(kecepatan_motor_kiri, 60);</code>
29	<code> analogWrite(kecepatan motor_kanan, 255);</code>

30	
31	analogWrite(e1, 255);
32	analogWrite(e2, 0);
33	analogWrite(d1, 0);
34	analogWrite(d2, 255);
35	}
36	}
37	if(kiri <=20 && kanan>20 && depan <=8) right();
38	
39	if(kiri >20 && kanan>20 && depan <=8) right();
40	
41	if(kanan <=20 && kiri>20 && depan <=8) left();
42	
43	if(kanan<=20 && kiri<=20 && depan<=8) back();

Pada tabel 5.5 bagian baris 1 sampai 34 program bermakud jika objek dinding lebih dari 8 cm maka robot akan mengendalikan diri agar tetap pada jalur lurus. Jika jarak dinding kiri kurang dari sama dengan 20 cm, kanan lebih dari 20 cm dan depan kurang dari sama dengan 8 cm maka robot akan berbelok ke kanan. Jika dinding kiri lebih dari 20 cm, kanan lebih dari 20 cm dan depan kurang dari sama dengan 8 cm maka robot akan berbelok ke kanan. Jika jarak dinding kanan kurang dari sama dengan 20 cm, , kanan lebih dari 20 cm dan depan kurang dari sama dengan 8 cm maka robot akan berbelok ke kiri untuk putarbalik jika semua sensor mendeteksi objek dinding kurang dari 8 cm.

5.2.3.5 Implementasi Penyimpanan ke Memori EEPROM dan *Maze Mapping Robot*

Tabel 5. 8 Penyimpanan ke Memori EEPROM dan Maze Mapping Robot

No	Source Code
1	void algorithm() {
2	int a;
3	FinishPin = digitalRead(10);
4	if (FinishPin == LOW) {
5	if (kiri <= 20 && kanan > 20 && depan <= 10) {
6	a = 1;
7	}
8	if (kiri > 20 && kanan > 20 && depan <= 10) {
9	a = 1;
10	}
11	if (kanan <= 20 && kiri > 20 && depan <= 10) {
12	a = 2;
13	}
14	if (kanan <= 20 && kiri <= 20 && depan <= 10) {
15	a = 3;
16	}
17	if (kanan <= 20 && kiri <= 20 && depan > 10) {
18	a = 0;
19	}


```

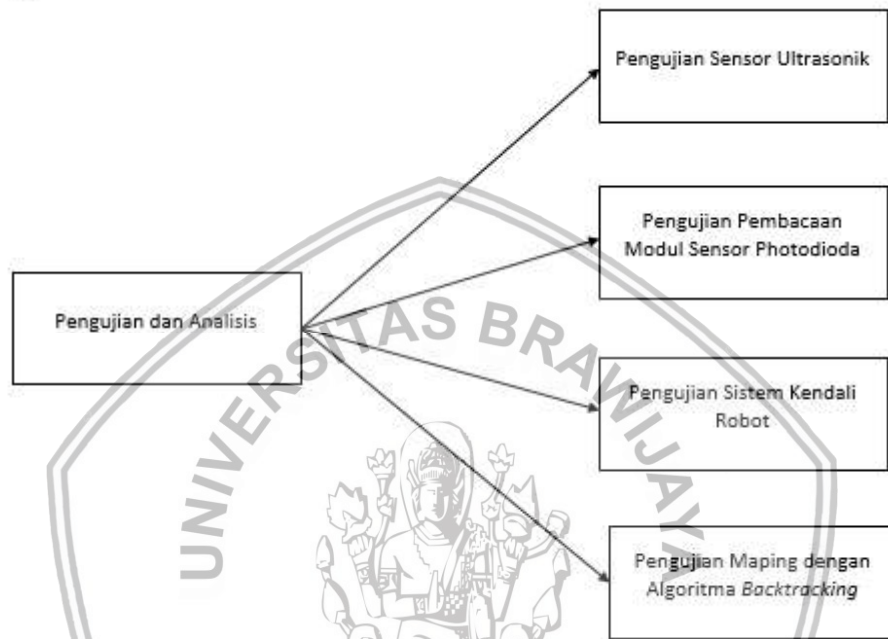
20
21     for (int x = 0; x <= 26; x++) {
22         if (EEPROM.read(x) == 0) {
23             delay(1000);
24             EEPROM.write(x, a);
25             Serial.print(x);
26             Serial.println(EEPROM.read(x));
27         }
28         else{
29             Serial.print(x);
30             Serial.println(EEPROM.read(x));
31         }
32     }

```

Pada program di atas jika sensor Photodiode mendeteksi warna gelap =LOW maka program akan menulis nilai di variabel array value pada EEPROM. Nilai yang ditulis adalah jika depan kurang dari 20cm kiri kurang dari 20cm dan kanan lebih dari 20 cm maka robot akan belok kanan yang ditandai dengan angka 3. Sedangkan untuk kanan kurang dari 20cm kiri lebih dari 20cm dan depan lebih dari 20cm maka robot akan lurus yang ditandai dengan angka 0. Serta jika semua sensor kurang dari 20cm maka robot akan putar balik yang ditandai dengan angka 3. Jika belok kiri robot akan ditandai dengan angka 2. Dari kode program di atas maka setiap robot melewati garis hitam robot akan membaca setiap sensor dan dari pembacaan sensor tersebut robot akan menandai daerah yang telah dilewati jika maju ditandai dengan angka 0, belok kanan ditandai dengan angka 1, belok kiri dengan angka 2 dan jalan buntu dengan angka 3. Jika robot tidak membaca garis hitam maka robot akan menampilkan langsung data yang ada pada memori EEPROM.

BAB VI PENGUJIAN DAN ANALISIS

Pada bab ini pembahasan pengujian dan analisis dari pengujian sensor Ultrasonik, sensor Photodiode, Sistem kendali dan Algoritme *backtracking*.



Gambar 6. 1 Grafik Pengujian dan Analisis

6.1 Pengujian Pembacaan Sensor Ultrasonik HC-SR04

Pada pengujian sensor Ultrasonik dengan melakukan pengujian jarak yang berbeda dengan menghitung akurasi dari jarak sensor ke objek yang dijadikan bahan pengujian.

6.1.1 Tujuan Pengujian

Untuk mengetahui keakuratan sensor Ultrasonik dan semakin jauh jarak sensor ke objek, bagaimana akurasi dan tampilan persentase keberhasilan

6.1.2 Prosedur Pengujian

Pada Gambar 6.2 prosedur pengujian sensor Ultrasonik dengan objek dinding. Pengujian menguji keakuratan dalam menghitung jarak yang asalnya dari waktu pantulan gelombang sensor dan diterimanya kembali



Gambar 6. 2 Pengujian Sensor Ultrasonik

1. Menghubungkan Robot dengan komputer atau laptop menggunakan kabel USB
2. Buka *sourcecode* yang disiapkan untuk pengujian sensor Ultrasonik. Da *compile Sourcecode* tersebut ke dalam Mikrokontroler Arduino Nano.
3. Siapkan suatu objek datar yang fungsinya untuk media pemantulan gelombang dari sensor Ultrasonik.
4. Melakukan perpindahan obek dengan semakinjauh objek
5. Mengumpulkan data hasil pengukuran data dari sensor Ultrasonik

6.1.3 Hasil dan Analisis Pengujian

Tabel 6. 1 Hasil Pengukuran Jarak Menggunakan Sensor Ultrasonik

Sensor kiri			Sensor depan			Sensor Kanan		
Jarak yang di ukur penggaris (cm)	Jarak yang di ukur sensor(cm)	Akur asi (%)	Jarak yang di ukur penggaris (cm)	Jarak yang di ukur sensor(cm)	Akur asi (%)	Jarak yang di ukur penggaris (cm)	Jarak yang di ukur sensor(cm)	Akur asi (%)
2	2	100	2	2	100	2	2	100
4	4	100	4	4	100	4	4	100
6	6	100	6	6	100	6	6	100
8	8	100	8	8	100	8	8	100
10	10	100	10	10	100	10	10	100

12	12	100	12	12	100	12	12	100
20	20	100	20	20	100	20	20	100
40	40	100	40	40	100	40	40	100
50	53	99.2	50	51	99.3	50	55	98
Rata-Rata Persentasi Akurasi sensor kiri		99	Rata-Rata Persentasi Akurasi sensor tengah		99	Rata-Rata Persentasi Akurasi sensor kanan		99

Sensor Ultrasonik bekerja berdasarkan perbandingan dari berapa waktu yang ditangkap setelah gelombang dipancarkan. Semakin jauh benda maka semakin jauh benda maka waktu pantulan akan semakin lama sedangkan jika semakin dekat benda maka pantulan akan semakin cepat. Jadi semakin jauh pantulan gelombang maka akurasi dari sensor pun akan berkurang. Hasil dapat dilihat pada tabel 6.1 pada jarak 50 cm akurasi sensor mulai berkurang.

6.2 Pengujian Pembacaan Modul Sensor Photodiode

Di sistem ini, sensor Photodiode digunakan untuk mendeteksi alas atau dasar labirin yang berwarna gelap atau terang. Pada pengujian Sensor Photodiode ini diperlukan adanya objek yang berwarna terang dan gelap untuk melihat dari hasil pembacaan sensor Photodiode.

6.2.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari modul sensor Ultrasonik dalam mendeteksi warna yang berwarna gelap dan terang.

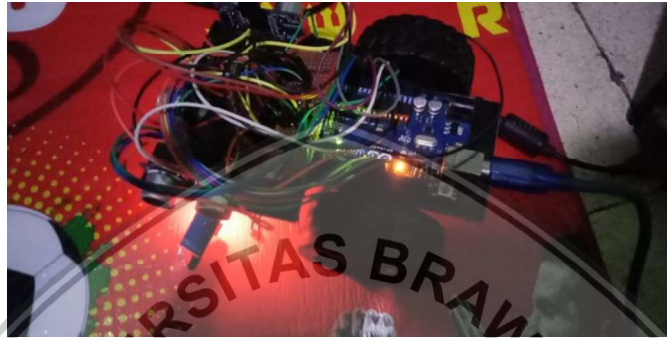
6.2.2 Prosedur Pengujian

Untuk menguji tingkat keakuratan dari modul sensor Photodiode akan dilakukan percobaan untuk mendeteksi objek yang berwarna. Pengujian menggunakan program untuk modul sensor Photodiode yang digunakan untuk mendeteksi warna gelap dan terang dan menampilkannya pada Serial monitor. Prosedur pengujian sensor Photodiode sebagai berikut:

1. Menghubungkan prototipe alat dengan PC menggunakan kabel USB dan LED untuk pencahayaan yang digunakan untuk sumber cahaya untuk pengujian Modul Sensor Photodiode.
2. Membuka *source code* yang dibuat untuk mendeteksi warna menggunakan sensor Photodiode. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.

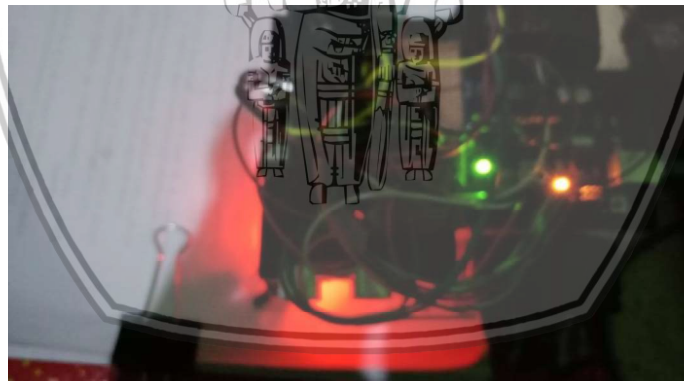
3. Memberi suatu objek berwarna yang berbeda-beda dan mengamati hasil yang ditampilkan pada serial monitor.
4. Membandingkan apakah keterangan yang ditampilkan pada serial monitor Arduino IDE.

Berikut pada Gambar 6.3 melakukan pengujian pada karpet warna merah, Gambar 6.4 melakukan pengujian pada kertas warna putih, Gambar 6.5 melakukan pengujian pada kain warna hitam.



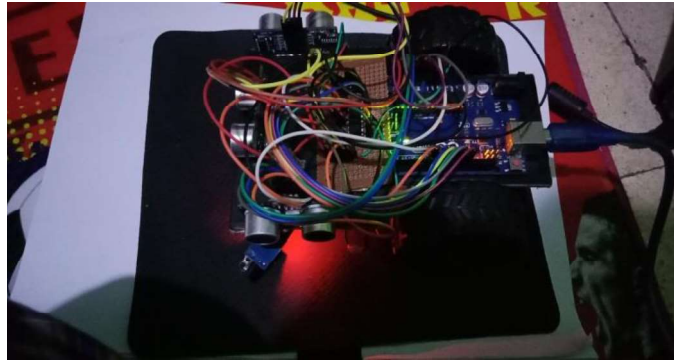
Gambar 6. 3 Pegujian Pada Warna Merah

Gambar 6.4 pada saat pengujian sensor photodiode pada warna putih. Pengujian menggunakan media kertas putih sebagai dasar robot.



Gambar 6. 4 Pengujian Pada Warna Putih

Gambar 6.5 pada saat pengujian sensor photodiode pada warna hitam. Pengujian menggunakan media kain hitam sebagai dasar robot.



Gambar 6. 5 Pengujian Pada Warna Hitam

6.2.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari sensor Photodiode yang digunakan dalam sistem ditampilkan pada Tabel 6.2.

Tabel 6.2 Hasil Pengujian Sensor Photodiode

Warna Objek	Pengujian Ke-	Pembacaan Sensor Photodiode	Akurasi
Merah	1	Warna Terang	100%
	2	Warna Terang	
	3	Warna Terang	
Putih	1	Warna Terang	100%
	2	Warna Terang	
	3	Warna Terang	
Hitam	1	Warna Gelap	100%
	2	Warna Gelap	
	3	Warna Gelap	
Coklat	1	Warna Terang	100%
	2	Warna Terang	
	3	Warna Terang	

Berdasarkan analisis Tabel 6.3 Modul sensor Photodiode ada beberapa warna yang di uji yaitu merah, putih, hitam dan coklat. Pada warna- warna tersebut hampir semua dapat memantulkan cahaya yang dapat ditangkap kembali oleh sensor Photodiode kecuali warna hitam. Karena warna hitam tidak dapat memantulkan cahaya yang dipancarkan LED.

6.3 Pengujian Sistem Kendali Robot

Di sistem ini, L298N Driver Motor digunakan untuk menentukan arah putaran dari kedua Motor DC. Pada pengujian L298N Driver Motor dilakukan uji coba untuk robot berbelok kiri, belok kanan, putar balik, dan berhenti.

6.3.1 Tujuan Pengujian

Untuk mengetahui tingkat keakuratan dari L298N Driver Motor saat melakukan belok kiri, belok kanan, putar balik dan berhenti.

6.3.2 Prosedur Pengujian

1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB.
2. Membuka *source code* yang dibuat untuk L298N Driver Motor. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
3. Membuat kondisi L298N Driver Motor untuk berbelok kiri, belok kanan, putar balik dan berhenti.
4. Mengamati pergerakan dari Motor DC kiri dan kanan.

Berikut Gambar 6.6 robot melakukan pengujian di dalam *maze*.



Gambar 6. 6 Pengujian Pergerakan Robot

6.3.3 Hasil dan Analisis Pengujian

Hasil dari pengujian tingkat akurasi keberhasilan dari Sistem Kendali Robot yang digunakan dalam sistem ditampilkan pada Tabel 6.3.

Tabel 6. 3 Hasil Pengujian Pergerakan Robot.

Pergerakan	Percobaan	Keterangan	% Keberhasilan
Belok Kanan	1	Berhasil	70%
	2	Tidak Berhasil	
	3	Tidak Berhasil	
	4	Berhasil	
	5	Berhasil	
Belok kiri	1	Berhasil	50%
	2	Berhasil	
	3	Tidak Berhasil	
	4	Tidak Berhasil	
	5	Berhasil	
Putar Balik	1	Tidak Berhasil	40%
	2	Tidak Berhasil	

	3	Tidak Berhasil	
	4	Berhasil	
	5	Berhasil	

Untuk mengetahui tingkat keberhasilan dan kehandalan sistem kendali robot yang digunakan, maka dilakukan pengujian terhadap pergerakan robot. Gerakan-gerakan yang sering dilakukan oleh robot seperti maju lurus, putar kanan, putar kiri, putar kiri. Sama halnya ketika robot melakukan pergerakan berputar penggunaan motor driver berperan sangat penting agar ketepatan berputar selalu terpenuhi. Untuk membuktikan keberhasilan pergerakan robot, maka dilakukan pengujian sebanyak 5 kali. Pada saat Putar balik persentase paling kecil dikarenakan robot sering sekali menabrak dinding yang ukurannya kecil untuk melakukan putar balik dengan mulus.

6.4 Pengujian Mapping dengan Algoritme *Backtracking*

Dalam pengujian Algoritme ini, robot dicoba untuk melakukan tugas yang sebenarnya. Tujuannya adalah apakah Algoritme yang digunakan bisa diterapkan atau tidak.

6.4.1 Tujuan Pengujian

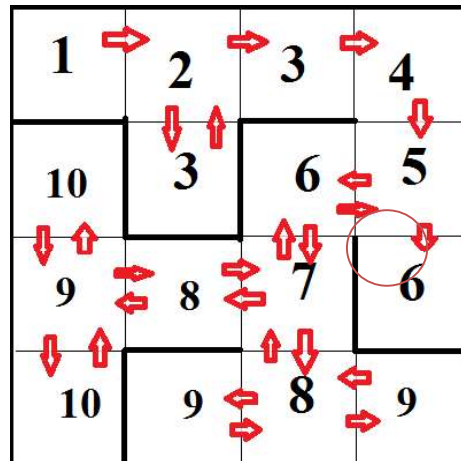
Untuk mengetahui apakah Algoritme Mapping dapat melakukan penandaan pada setiap *Grid* maze yang telah dilewati robot.

6.4.2 Prosedur Pengujian

1. Menghubungkan prototipe alat dengan komputer atau laptop menggunakan kabel USB.
2. Membuka *source code* yang dibuat untuk Algoritma Mapping. Setelah itu melakukan *compile* dan mengunggah *source code* program ke Arduino Nano.
3. Memulai Pengujian dari *Start Maze*, Siapkan Serial monitor Untuk mengamati data yang masuk ke *Grid* yang telah dilewati robot.
4. Amati Sampai selesai dan amati data yang masuk ke dalam serial monitor.

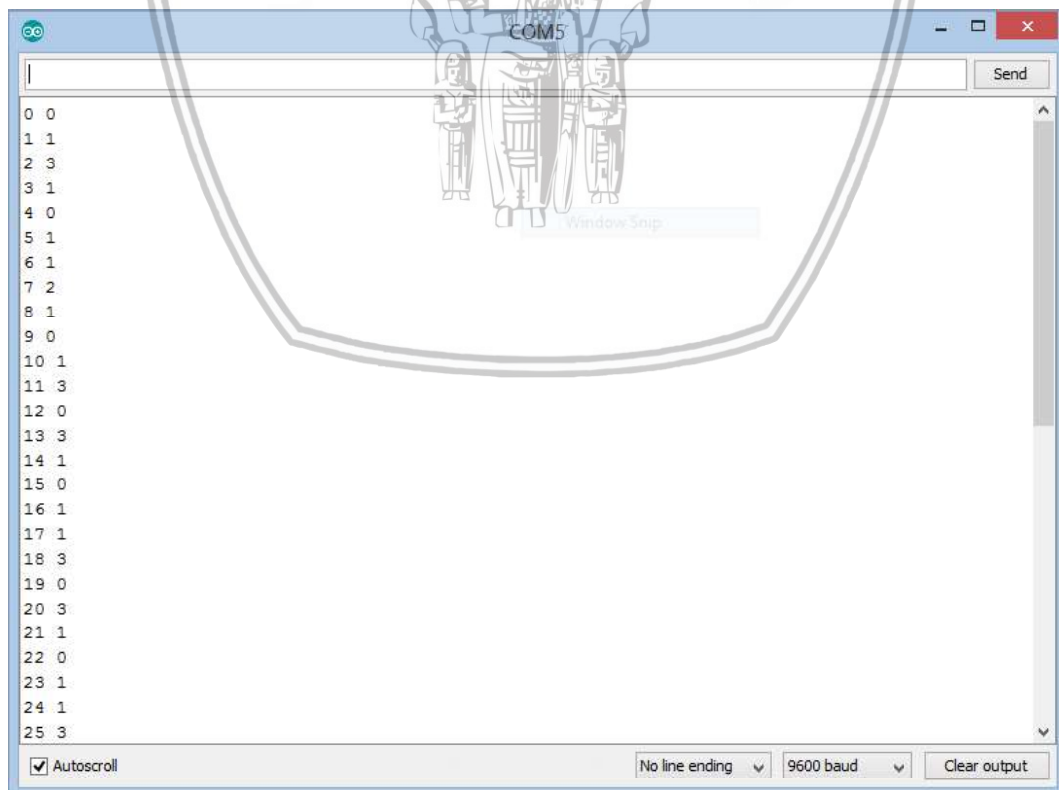
6.4.3 Hasil dan Analisis Pengujian

Hasil dari pengujian Algoritme Mapping dapat dilihat dari Gambar 6.7. yang menggambarkan arah jalan robot dan penomoran setiap *Grid* agar *Grid* yang sudah dilewati tidak dapat dilewati kembali.



Gambar 6. 7 Arah Jalannya Robot dan Hasil Mapping

Dari gambar di atas dapat dilihat start teadapat pada nomor 1 dan robot brhenti pada nomor 6. Setelah melewati salah satu *Grid maze* robot memberi tanda berupa nomor yang tertera pada gambar yang artinya robot telah melewati *Grid* tersebut. Jika robot melewati *Grid* yang sama untuk mencari *Grid* yang kosong *Grid* yang pernah dilewati tidak dapat di beri tanda lagi agar tidak terjadi perulangan. Gambar 6.8 berikut merupakan tampilan data array yang tersimpan pada memori EEPROM.



Gambar 6. 8 Tampilan Data Hasil Mapping pada Serial Monitor

Pada tampilan serial mornitor pada Gambar 6.8 merupakan hasil dari pembacaan data yang ada pada memori EEPROM. Pada data tersebut menerangkan setiap langkah robot atau setiap robot melewati garis hitam maka robot akan menandai wilayahnya dengan tanda jika belok kanan angka 1 jika belok kiri angka 2 jika putar balik angka 3 dan jika lurus 0. Dari data tersebut terdapat 25 langkah yang merupakan data setiap *grid* yang telah dilewati dari data tersebut dapat dilihat bahwa dari *grid maze* ke 1 sampai ke 6 (jalan buntu) robot akan terus memutar *maze* sampai semua *grid* terlewati.



BAB VII PENUTUP

Bab 7 ini berisi kesimpulan yang didasarkan dari pengujian dan analisis yang dilakukan selama proses penelitian dan saran yang berisi hal-hal yang diperlukan dalam melakukan pengembangan untuk topik skripsi selanjutnya.

7.1 Kesimpulan

Berdasarkan pengujian dan Analisis yang dilakukan terhadap tugas akhir ini maka didapatkan kesimpulan sebagai berikut :

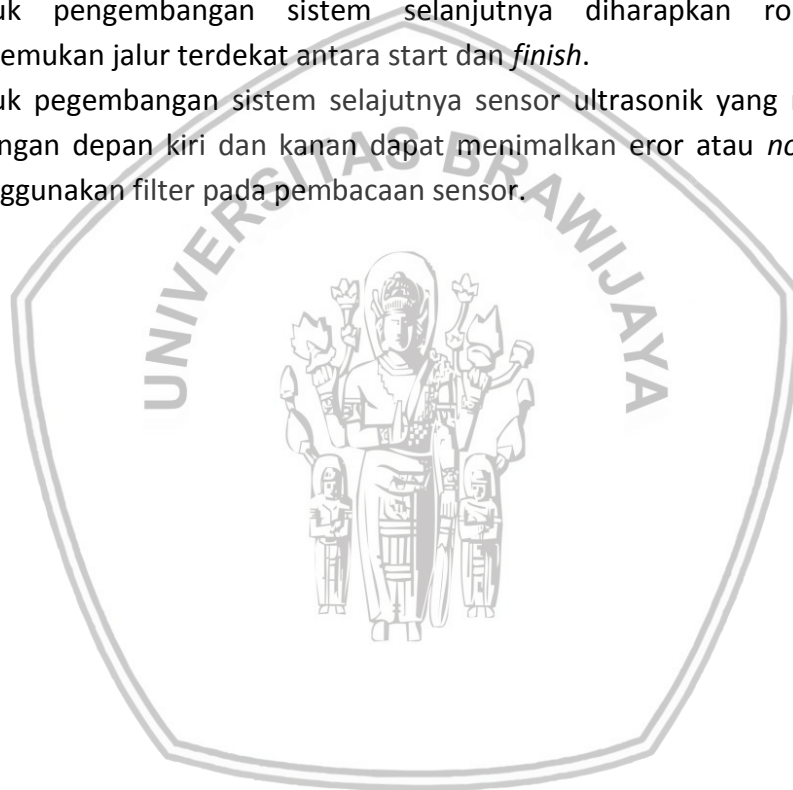
- 1) Robot mendeteksi jarak antar dinding dengan robot dengan menggunakan sensor ultrasonik HC SR04 yang letaknya dibagian depan, samping kanan dan samping kiri. Peletakan sensor di buat berdiri tegak agar dapat mendeteksi dinding dengan baik. Dari hasil pengujian sensor ultrasonik sebanyak 7 kali dengan jarak yang berbeda-beda hasilnya semakin jauh keberadaan sensor dengan objek maka semakin tidak akurat juga hasilnya. Sedangkan untuk mendeteksi garis pada setiap batas node pada dasar labirin menggunakan modul sensor photodiode yang pengujiannya menggunakan 3 warna berbeda yaitu merah, putih dan hitam. Setiap warna dilakukan pengujian sebanyak 3 kali. Hasilnya hanya warna hitam yang tidak dapat memantulkan cahaya yang dapat diterima oleh sensor photodiode. Persentase keakuratan dari sensor mencapai 100% jika jarak yang dideteksi kurang dari 50 cm. Jika lebih dari 50 cm keakuratan berkurang sampai 99%.
- 2) Untuk mengenali setiap node robot menggunakan sensor photodiode. Setiap node pada labirin ditandai dengan garis hitam yang membatasi antara satu node dengan yang lainnya. Dari garis hitam tersebut robot dapat mengenali setiap node dengan bantuan sensor photodiode. Setiap sensor membaca garis hitam maka nilai dari sensor akan tersimpan di dalam Array yang terdapat pada memori EEPROM yang nilainya akan terus tersimpan meskipun robot mati nilai pada memori EEPROM akan terus ada. Dari nilai yang dimasukkan ke memori EEPROM robot dapat mengenali setiap node yang telah dilewati robot. Keakuratan dari pengujian sensor mencapai 100% pada setiap warna yang diuji.
- 3) Implementasi Algoritme *Backtracking* robot dapat mengenali setiap node pada maze dengan masukan dari nilai sensor photodiode yang tersimpan pada Array 2 dimensi yang tersimpan pada memori EEPROM. Jika robot telah melewati sebuah node maka nilai akan tersimpan pada array di memori EEPROM. Dengan metode dasar *follow right* pada robot maka robot akan meng eksplor maze dengan memprioritaskan belok kanan. Robot akan terus

bergerak jika ada array yang masih kosong. Agar robot tidak dapat melewati jalur yang sama maka nilai dari sensor photodiode akan masuk ke array dan jika array itu sudah penuh maka robot otomatis akan berhenti bergerak. Ketepatan dalam sensor mendeteksi garis hitam dan datanya langsung tersimpan dalam memori EEPROM mencapai 100%

7.2 Saran

Berdasarkan pengujian dan Analisis yang sudah dilakukan terhadap tugas akhir ini maka ada beberapa saran yang dapat digunakan untuk pengembangan sistem ini atau implementasi dengan sistem yang lain sebagai berikut:

- 1) Untuk pengembangan sistem selanjutnya diharapkan robot dapat menemukan jalur terdekat antara start dan *finish*.
- 2) Untuk pengembangan sistem selanjutnya sensor ultrasonik yang mendeteksi halangan depan kiri dan kanan dapat meminimalkan eror atau *noise* dengan menggunakan filter pada pembacaan sensor.



DAFTAR PUSTAKA

- Adjie, S. (2015). *ARDUINO IDE*. Retrieved from Pengenalan Arduino: www.Septaadjie.web.id
- Ahyar, S. (2011). Pelacak Jarak Untuk Navigasi Gerak Robot.
- Anak Kendali. (2018). *Pembacaan dan Penulisan dalam EEPROM pada Arduino Uno*. Retrieved from Anak Kendali: www.anakkendali.com
- Depok Instruments. (2016, Februari 23). *HC-SR04 (Ultrasonic Sensor)*. Diambil kembali dari DEPOK INSTRUMENTS: <https://depokinstruments.com/2016/02/23/hc-sr04-ultrasonic-sensor/>
- Dong W, K., Ty A, L., & Steven A, V. (2013). Autonomus Multi-mobile Robot System: Simulation and Implementation Using Fuzzy Logic. 1.
- Ecadio. (2018). *Mengenal Arduino Uno R3*. Diambil kembali dari Toko Online Arduino: <http://ecadio.com/mengenal-dan-belajar-arduino-uno-r3>
- ELEKTRONIC ART. (2013). *Driver Motor DC L293D*. Diambil kembali dari ELEKTRONIKA DASAR: <http://elektronika-dasar.web.id/driver-motor-dc-l293d/>
- Fauzi, I. (2017, September 19). *Autonomous Robotics Surgery*. Diambil kembali dari Unit Robotika ITB: <http://robotika.unit.itb.ac.id/main/1173-autonomous-robotics-surgery.html>
- Guntomo. (2011). *Software Arduino*. Retrieved from Arduino IDE: www.Guntomolecture.web.id
- Hendriawan, A., & Akbar, R. (2017). Penyelesaian Jalur Terpendek dengan menggunakan Algoritme Flood Fill pada Line Maze . 1.
- Manika, A. (2016, November 25). *Kenapa Baterai pada Akhirnya Mati?* Diambil kembali dari blajar: <http://blajar.org/pertanyaan/10814/>
- Matteo, D. (2017, Agustus 15). *Line maze solving robot*. Diambil kembali dari <https://forum.arduino.cc/index.php?topic=495033.0>
- Rame, I. (2013). *MAKALAH ROBOT LINE FOLOWER*. Tembarak: SMK NEGERI TEMBARAK.

Teknik Elektronika. (2017). *Pengertian Motor DC dan Prinsip Kerjanya*. Retrieved from Teknik Elektronika: <http://teknikelektronika.com/pengertian-motor-dc-prinsip-kerja-dc-motor/>

Tirian, & Gelu, O. (2015). *Maze Solving Mobile Robot*.

Tjiharjadi, Semuil, & Setiawan, E. (2016). *Design and Implementation of a Path Finding Robot Using Floodfill Algorithm*.

